7-28-2010

Improved approximations for flows of thin films

Christine Elizabeth Lowry

Improved Approximations for Flows of Thin Films

by

Christine Elizabeth Lowry

A thesis submitted in partial fulfillment of the requirements for the
Master of Science
in
Mechanical Engineering

Approved by:

Dr. Kenneth Ruschak _____
Department of Chemical and Biomedical Engineering
Thesis Advisor

Dr. Steven Weinstein _____
Department of Chemical and Biomedical Engineering
Thesis Committee

Dr. Risa Robinson _____
Department of Mechanical Engineering
Thesis Committee

Dr. Kathleen Lamkin-Kennard _____
Department of Mechanical Engineering
Thesis Committee

Dr. Alan Nye _____
Department of Mechanical Engineering
Department Representative

Department of Mechanical Engineering
Kate Gleason College of Engineering

Rochester Institute of Technology
Rochester NY, 14623

July 28, 2010

Permission for Thesis Reproduction

I, Christine Elizabeth Lowry, hereby grant permission to the Wallace Memorial Library of Rochester Institute of Technology to reproduce my thesis entitled *Improved Approximations for Thin Film Flows* in whole or in part. Any reproduction will not be for commercial use or profit.

Christine Elizabeth Lowry	July 28, 2010

Abstract

Film flow equations are simplified equations for modeling the flow of thin liquid films. They are ordinary differential equations in terms of the film thickness. Typically, the boundary-layer approximation to the Navier-Stokes equation is employed, a velocity profile is assumed, and conservation of momentum then yields a film equation. Surface tension is usually important in which case the film equations are third order. Existing film equations are adequate when the substrate is not moving and when the substrate is moving in the absence of inertial effects. These equations are deficient in the important case of a moving substrate when inertia is included and the film connects with a reservoir of liquid that is substantially hydrostatic. In that case, the inertial terms of the film equations do not die off as the film thickens, unlike the viscous terms or the inertial terms when the wall is stationary, and so a hydrostatic reservoir is not described. The main goals of this thesis are to explore the cause of the deficiency and, if possible, propose a remedy.

To test the starting hypothesis that the assumption of a parabolic velocity profile is the cause of the deficiency, three studies were conducted. First, the full set of boundary-layer equations was solved exactly, without assuming a velocity profile, for the linearized case where the film thickness is close to its value far downstream. The velocity profile was found to be either parabolic, when inertia was neglected, or very close to parabolic when inertia was included. Second, two model problems with fixed boundaries were solved using the commercial code Fluent. Parabolic or near parabolic velocity profiles were also found. Third, complete free boundary calculations were obtained from collaborators having their own CFD codes for the flow called slot coating. These results again supported the use of a parabolic velocity profile as a good approximation. Furthermore, film profiles based on the parabolic film equation matched the CFD profiles well when inertial terms were small to moderate.

With a parabolic velocity profile established as appropriate, an approach other than introducing a non-parabolic velocity profile was indicated. It was recognized that

evaluating the inertial terms using average velocity, as traditionally done for flow in conduits, while continuing to evaluate the viscous terms using the parabolic velocity profile, gives rise to a film equation where the inertial terms die off as the film thickness increases as desired. The downstream asymptotic behavior of this plug-parabolic film equation was shown to be identical to that of the parabolic film equation. An unexpected but additionally desired behavior was exhibited: film profiles could be computed when inertial terms were dominant. As inertia increased, the profiles exhibited curvature relaxation, unlike the parabolic and other known film equations. Curvature relaxation due to inertia is an important phenomenon in coating, resulting in the ability to coat thinner as speed is increased and the disappearance of the instability called ribbing.

Predictions of coating thickness for the dip coating process, where a film is withdrawn from a pool of liquid, were made using the plug-parabola film equation. Comparisons with experimental and CFD data were limited because of a lack of results when inertia is important. Comparisons with experimental and computational data for slot coating show the correct qualitative behavior but indicate an early onset of meniscus relaxation. It may be necessary to modify the plug-parabola film equation, perhaps by blending it with the parabolic film equation, to delay the onset of curvature relaxation.

Acknowledgements

I would like to thank all of the following people for their support and guidance during my thesis. Ken, I could not have completed my thesis without your patience, willingness to teach and expertise…thank you. Steve, your guidance and support throughout the process of my thesis was invaluable. Last but not least my family: my fiancé Alex, parents and siblings. Thank you for always believing I could do it even when I did not believe.

Table of Contents

Appendices

List of Figures

Chapter One

Background and Goal of Thesis

The laminar flow of thin liquid films has many applications from paint application to the manufacture of electronic displays. Film flows occur as components of heat and mass transfer and other unit operations, examples being drying and mass transfer columns. Coating processes are an important application of laminar thin film flows. Coating technology is used to manufacture diverse products including papers, construction materials such as flooring and roofing shingles, layered electronic components, packaging materials, and optical films [1].

Currently, the numeric solution to the full set of governing equations, the Navier-Stokes equations, with a liquid/air interface, is challenging even for current numerical methods [2, 3]. The liquid/air interfaces are called free surfaces (FS) and are shown in figure 1.1 a schematic of a slot coater. In a slot coater, liquid issues from the slot of a coating die and transfers across a small gap onto a moving web. These solutions are computationally intensive free-surface problems and are inherently highly non-linear. This is because the position of the liquid/air interface is not known at the start of the problem but is determined as part of the solution. Surface tension effects are usually important in thin films.

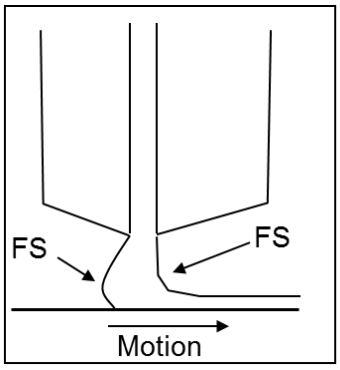

Figure 1.1: Depiction of the free surfaces in coating flows

Simplifying approximations to the Navier-Stokes equations and free surface boundary conditions can be made because the film is thin and variations in the direction of flow are gradual. This is the traditional and still dominant approach to free surface problems. The boundary-layer approximation to the Navier-Stokes equation can be made because the streamlines in thin film flows are nearly parallel. These simplified equations are easier to solve, use and extend. Applications such as the design of coating processes are easier and less demanding on time.

Film flow equations are usually solved by integrating across the film and introducing a velocity profile. Often the assumed velocity profile is a parabola, the exact profile when streamlines are parallel. The pressure in the film is determined by surface tension and the curvature of the liquid/air interface. In the resulting film equation, the film thickness is the only unknown. This film equation is a third order nonlinear ordinary differential equation that is straightforward to solve numerically. This process for creating film flow equations is detailed within Chapter 3 and is visually shown in figure 3.1.

The simplest film flow equation is the Landau-Levich equation shown in dimensionless form in equation 1.1 [3]. It neglects inertia and gravity in the thin film. The term on the left derives from the capillary pressure gradient and the term on the right from the viscous flow term. In equation 1.1, h is the film thickness and x is the direction horizontal to the film. It is clear that as the film thickness becomes large the viscous flow term becomes negligible.

$$\frac{d^3h}{dx^3} = \frac{3(1-h)}{h^3} \tag{1.1}$$

In the case where gravity and inertia have been retained, but surface tension has been neglected, the film flow equations based on a parabolic velocity profile fail to describe the hydrostatic reservoir [4]. The inertial term in this film flow equation slowly grows as the film thickens instead of dying off. Therefore, the film equation fails to describe the hydrostatic reservoir as desired [4]. However, when the film flow equation based on a

parabolic velocity profile is solved for the flow of the film down a stationary wall into a reservoir the deficiency of the film flow equation is not seen [5]. With a stationary wall, with inertia and gravity in the film region retained, both the viscous and inertial terms die off as the film thickens so that a hydrostatic reservoir is properly described. Therefore, the combination of inertia and a moving wall produces the incompatibility of the parabolic film flow equation to describe a hydrostatic reservoir.

The purpose of this work is to find the cause of the deficiency of the film flow equations with a moving wall and to create an improved film flow equation if possible. The starting hypothesis for the deficiency is the assumed form for the velocity profile. Therefore, in order to find the correct velocity profile, two representative numerical problems will be solved using Fluent and the velocity profiles extracted. The full PDE solution to the boundary-layer approximation without assuming a velocity profile will be investigated as well. These problems will be used to show the effect of inertia on the velocity profile within the film and to see if there is a large deviation from a parabola. As a result, it is hoped that an improved film flow equation will be found which will shorten and aid the engineering design process for coating and other applications.

One test problem for this work will be the dip coating flow [3]. Dip coating is the process where a substrate is withdrawn from a reservoir of liquid to form a film. Figures 1.2 is an illustration of dip coating with a moving wall. This problem was chosen because it exhibits the deficiency of the film flow equations in use and because it has been extensively studied both experimentally and theoretically and because of its practical importance.

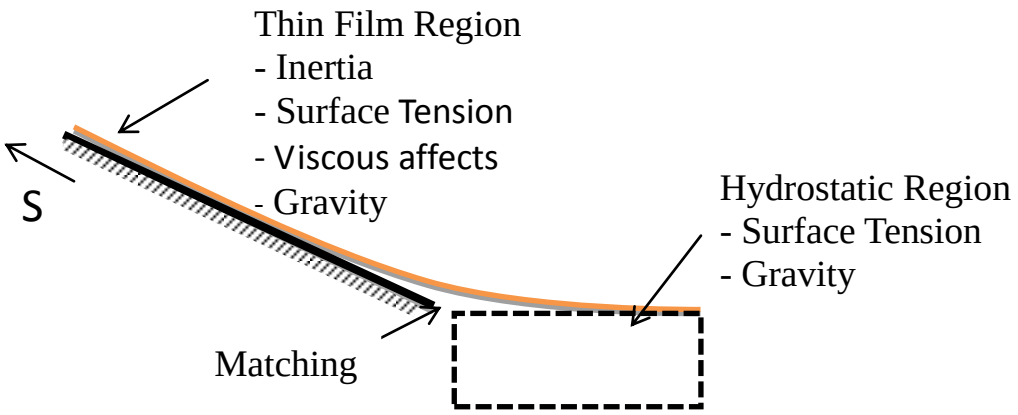

Figure 1.2: General dip coating process

Also, listed in figure 1.2 are the different regions of the flow and the important physical features for each. The matching region is where the transition from a thin film to a hydrostatic reservoir occurs. For consistency to describe the hydrostatic region, the viscous and inertial terms must become unimportant while the streamlines are still nearly parallel.

A second test problem which will be used is slot die coating. Whereas, in dip coating the flow rate is computed as part of the solution to the problem; in slot coating the flow rate is imposed usually by a pump. The slot within the die ends a short distance from the moving substrate which entrains the liquid to form a film. This is shown in figure 1.3. The die itself extends in and out of the page along the width of the substrate being coated. The results from this problem will help illustrate the effect of inertia on the film profile equations.

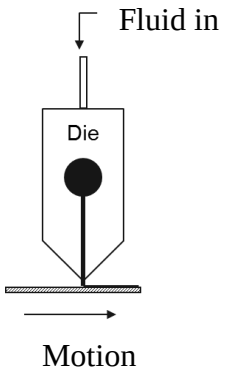

Figure 1.3: General slot die coater

Chapter Two

Literature Review

1. R. E. Hildebrand and J. A. Tallmadge, "Direct Evaluation of Withdrawal Equations," *AIChE Journal,* vol. 14, pp. 660-661, 1968.

The purpose of this paper was to describe a direct method not using iteration to evaluate withdrawal equations. The equations stated in this paper predict film thickness as a function of withdrawal speed under steady and continuous upward motion of a symmetric object through a free surface. This simplification of the complex withdrawal equations was examined for flat plates and cylinders.

This paper is noteworthy because it shows the process and steps taken to evaluated withdrawal equations in Newtonian fluids with different shapes. The approximations made and final equations are useful and create a simple way of solving or beginning to solve withdrawal equations.

This paper's strengths are that the withdrawal equations are solved for different applications and shapes such as flat plates and cylinders. One drawback to this paper is that it focuses on Newtonian fluids only and does not address how the equations might change for other fluids. The final withdrawal equations presented in this paper allow for a better physical understanding of what happens during a withdrawal process. The withdrawal equations for a flat plate and a cylinder involve similar physics to the problem addressed in this thesis.

2. R.P. Spiers, C. V. Subbaraman and W. L. Wilkinson, "Free coating of a Newtonian liquid onto a vertical surface," *Chemical Engineering Science,* vol. 29, pp. 389-396, 1974.

The vertical dip coating problem is analyzed. The objective of this paper was to predict the thickness of the film and the liquid flux as a function of time. The author assumed

that inertia would not affect the solution to the problem at low withdraw speeds from the pool. The authors included surface tension, gravity and viscous forces.

There was found to be good agreement between the new theoretical approach they found that included gravity and the other forces mentioned. However, the agreement between the data is good until a capillary number of 2. Through their work, they found that for capillary numbers greater than 2, the thickness becomes almost constant. The low speeds used were not identified. The theoretical solution found by the authors was compared to Landau-Levich.

This paper's strength appeared in the authors' experimental set-up. This set-up was shown to have a higher accuracy rate then previous experimental set-ups. The data that was collected experimentally was easily read off of the plots comparing it to Landau and Levich as well as Whit and Tallmadge. This paper also kept the viscous term of the normal stress in the problem. This paper also explains each region of the dip coating problem separately.

3. B. G. Higgins, W. J. Silliman, R. A. Brown, and L. E. Scriven, "Theory of Meniscus Shape In Film Flows. A Synthesis," *Industrial and Engineering Chemistry Fundamentals,* vol. 16, pp. 393-401, 1977.

The purpose of this paper was to demonstrate the different ways to solve free surface problems by the boundary conditions and the capillary number. The problem was solved using the Navier-Stokes continuity and momentum equations and three different approaches. The approaches used were the integral balances, differential approximation and perturbation techniques. These were used to solve three general coating problems--dip coating, surface leveling and rimming flow. The paper drew conclusions on how the different approaches in the literature are connected to the full numerical solution of the equations.

It was found that the basic equations of two-dimensional flows can be transformed into mathematical sets in a number of different ways. These different methods of simplifying the equations have different areas where they are valid.

This paper started with the fundamental test of film equations and then explained each part in turn. This paper's strength was in the clearness of the authors' insight into the different techniques that could be used if certain information is known. The flow of the equation solution as well as the logic used to eliminate certain terms was well documented.

4. K. J. Ruschak, "Flow of Falling Film into a Pool," *AIChE Journal,* vol. 24, pp. 705-709, 1978.

The objective of this paper was to show an easy method for simplifying and solving the governing equations for flow down a vertical wall. This paper also demonstrated that these equations can be generalized to non-vertical walls. The author discarded terms in the Navier-Stokes equations.

It also demonstrates that the simplified set of equations found can predict the aspects of the exiting flow. The paper finds that there is a simpler way to reduce the Navier-Stokes equations in order to solve this problem more efficiently.

This paper clearly defines the approach taken to simplify the governing equations of the film without linearizing the equations. Within the solution of the problem, the author showed that the viscous effects die off at approximately the same rate as the inertia term. Without linearizing the equations, the author keeps the transition zone a part of the solution to the problem.

This paper has important information that will help solve the film equations for the dip coating problem of this thesis; however, the wall is not moving in this case and is different from the thesis topic.

5. D. Marques, V. Costanza, and R. L. Cerro, "Dip Coating at Large Capillary Numbers: An Initial Value Problem," *Chemical Engineering Science,* vol. 33, pp. 87-93, 1978.

The objective of this paper was to determine the flow profiles in the thin film adhered to a solid surface that is removed from a liquid bath and to show that the maximum dimensionless flow rate attainable experimentally is dependent on the viscous, inertial and gravitational forces. It was found that during dip coating at large capillary numbers the inertial forces became more prevalent as opposed to the capillary forces. It was found that the equations of motion lost their ability to give information upstream. Due to this fact, it was shown that the problem became an initial value problem.

The problem was solved starting from the Landau-Levich equation and using finite-difference methods. Many different velocity profiles were tested and showed physically unacceptable behavior. This is significant because it shows that there are discrepancies associated with the Landau-Levich equation.

One of the strengths of this paper was that the mathematical approach was clear and the plots were completely explained and the reader could reproduce the author's work if necessary. A negative aspect of this paper was that it eliminated the negative velocities that appeared downstream. These terms were then used as new velocity profiles. The author did not focus on understating what caused these negative velocities. It is believed that this could have altered his results.

6. S. D. R. Wilson, "The drag-out problem in film coating theory," *Journal of Engineering Mathematics,* vol. 16, pp. 209-221, 1982.

The main objectives of this paper were to take the Landau-Levich result and show that it is an asymptotic solution valid for a capillary number that trends to zero. Currently, the Landau-Levich's approximant solution to the film thickness of a liquid film, when it is

drawn from a pool of liquid at a steady speed, is valid for small capillary numbers. Also, this paper hoped to find a correction term using method of matching.

Existing experimental data was used to show that the Landau-Levich solution is an overestimate of the film thickness. The correction factors for the Landau-Levich equation were found to have an affect on the correlation between the experimental data and the equation; however, the exact difference is unknown.

This paper's methodology makes sense in how the problem was approached and solved. The strength of this paper is that it includes the gravity term in the Landau-Levich equation. Often this term is neglected when solving the Landau-Levich equation. Within the paper, the author comments on the instabilities that arose from solving the problem; however, it is believed that he could have further explained and expanded the physics and how it affects the stability of the problem.

7. H. I. Andersson, "On Integral Method Predictions of Laminar Film Flow," *Chemical Engineering Science,* vol. 39, pp. 1005-1010, 1984.

The purpose of this paper was to derive an implicit analytical solution for the film thickness down an incline plane. The integral analysis was improved for the flow in the entrance region. Streamwise pressure gradient was taken into account while solving the problem. Gravitational and viscous effects were accounted for in the analysis. This paper also shows a pervious error in another paper.

The solution to the integral analysis showed that the solution depends on film, Reynolds number and angle of inclination. There was a reduction in the entrance length when the streamwise pressure was included. The analytical solution was compared with experimental data and was found to have reasonable agreement.

This paper solved a thin film flow down an incline plane. A strength of the paper is that the author included the y component of gravity in the analysis. A weakness of the paper is that the fluid does not return to a reservoir or pool.

8. P. Tanguy, M. Fortin, and L. Choplin, "Finite Element Simulation of Dip Coating, I: Newtonian Fluids," *International Journal For Numerical methods In Fluids*, vol. 4, pp. 441-457, 1984.

This paper focused on using finite element simulation on dip coating. It was hoped that the use of this simulation would provide information about the thickness of the film at the top of the plate. This simulation used discontinuous pressure elements and discretization of the continuum. The simulation computed the flow field from the natural boundary conditions that also allowed for a free surface. A stagnation point and the free surface were included in the simulation as well as surface tension, viscosity and gravity. The software used for this paper also allowed for the creation of the air-liquid interface and has the meniscus results from the equilibrium between viscous forces and surface tension.

By using published data on dip coating, the simulation showed good comparison with the experimental data for dip coating. It was found that the finite element simulation compared well with the published data. This paper shows how finite elements can be implemented to solve dip coating problems with an acceptable accuracy rate.

A strength of this paper was that it included circulation of the flow and the free surface aspect of the dip coating process. Also, the problem was solved for Newtonian and non-Newtonian fluids as well. This paper approached the dip coating problem as a vertical wire being withdrawn from a pool. It was assumed that the flow was steady, laminar and incompressible to order to find a simplified approach to solving the finite element dip coating problem.

9. K. S. Kheshgi, S. F. Kistler, and L. E. Scriven, "Rising and Falling Film Flows: Viewed From a First-Order Approximation," *Chemical Engineering Science,* vol. 47, pp. 683-694, 1992.

This paper analyzed a steady flow on a vertical solid surface. A rectangular approximation at low capillary numbers was used. Two solutions were found to the equations: first, that the bath meters the rising film flow and final film thickness, and the second that the final thickness of a rising or falling film is controlled by the conditions above the bath.

It was shown that an ordinary differential equation for steady rising or falling flow can be derived from the Navier-Stokes equations. Also, the bath configuration controlled the final film thickness; in contrast, the bath did not control the final film thickness for the falling films.

A strength of this paper is that the author establishes that a first order approximation shows no internal layers in the falling film. If the author had left in inertia there would be a change in the results. Another strength of this paper is that it recognizes that there are two flow regimes and that they are linked by the transition zone. The author focuses on gravity driven flows and does not include a scenario when the wall is moving.

This thesis is centered on the same basic dip coating problem allowing for me to have more exposure to different solution methods and approaches to these types of problems.

10. S. J. Weinstein and K. J. Ruschak, "Dip coating on a planar non-vertical substrate in the limit of negligible surface tension," *Chemical Engineering Science,* vol. 56, pp. 4957-4969, 2001.

The objective of this paper was to determine the range of possible dip coating solutions with negligible surface tension. Another objective was to investigate portions of the solution where the critical point does not determine the final film thickness. Dip coating was analyzed on a planer non-vertical surface. This is a self-metered coating which means that the film thickness is determined by the nature. The final film thickness is determined by the speed of the withdrawal, angle of the web and the fluid properties. The high-averaged approach was used to solve the non-vertical dip coating problem.

It was shown that there is a singularity point from the approximate steady-state equations governing the air-liquid interface and that the final film thickness can be determined by eliminating this singularity point. Also, there can be only two possible uniform film thickness solutions for a given volumetric flow rate; one superficial and one subcritical upstream and downstream of the singularity point, respectively. Metering of the flow and the final film thickness downstream was due to the critical point.

This paper has many strengths and the problem is well defined. It is easy to follow the steps taken and the thought process of how the problem was solved. The paper clearly defines and shows the equations and assumptions used to solve the dip coating on a planar non-vertical substrate problem. The importance of the problem and how it is applicable to other aspects of coating was made clear in this paper.

This dip coating problem is the same test case that will be solved. The main difference is that surface tension will be included.

11. P. Groenveld, "High capillary number withdrawal from viscous Newtonian liquids by flat plates," *Chemical Engineering Science,* vol. 25, pp. 33-40, 1969.

This paper focused on film thickness results at high capillary numbers from an experiment. A dip coating problem was used in the experiment. In order to measure the film thickness, a light absorption method was used. A circular glass plate and a colored liquid were used for measurement purposes. The experiment consisted of eight different fluid solutions: seven glycerol-water; and one sugar-syrup ranging from low to high viscosity. Each of these fluids was tested at a range of velocities, 0.00235 to .439 m/sec. Because of this range in speed, a variety of capillary numbers were able to be tested. This is one of the few papers that published the results from the experiment in table form. The dip coating problem will be used to support any new film equation that is found in this work because this work focuses on the effects of inertia on the film. This set of published data will be useful for validating any findings within this thesis.

12. B. Jin, A. Acrivos, and A. Munch, "The drag-out problem in film coating," *Physics of Fluids,* vol. 17, pp. 103603-12, 2005.

This paper solves a dip coating problem for the asymptotic film thickness using the full steady-state Navier stokes equations. The solutions to this free surface problem were found using FIDAP, a computational fluid dynamics software. A limit was found for Bond number that agreed with previous work.

This paper's numerical results for the meniscus shape at low capillary numbers will be used to verify the effectiveness of the proposed solution for this thesis when inertial affects are small.

13. M. Carvalho and H. Kheshgi, "Low-Flow Limit in Slot-Coating: Theory and Experiments," *AIChE*, vol. 46, pp. 1907-1917, 2000

The goal of this paper was to come up with a theoretical way to predict the coating window for a slot coater. The theory was compared to experimental data that measured the meniscus shape as well as the critical gap thickness between the coater and the substrate. The theoretical results found did not agree with the experimental results; however, the experimental results showed the relaxing and tightening of the meniscus shape as the capillary number is increased. The paper first describes that as the capillary number increases, the meniscus shape relaxes; however, there is a point where the meniscus shape begins to tighten as the capillary number increases. Another important result of this paper is that at faster web speeds thinner films can be coated.

This paper has a great deal of information about the effects of inertia on meniscus shape as well as information on the critical gap thickness for a slot coater. This information is useful for comparison of this thesis theoretical work with experimental work and will help illustrate the new proposed film equation effectiveness in predicting the effects of inertia on the meniscus shape.

Chapter 3

Film Equation Solutions

3.1 Introduction

In this chapter, the derivations of the appropriate film equations are provided. Figure 3.1 illustrates the derivation of these film equations from the Navier-Stokes equations. Figure 3.1 shows the motivation for creating the film equations. These equations are simplifications that are easier to solve than the full Navier-Stokes or boundary-layer approximation equations. The simplifying progression is from the Navier-Stokes equation to the parabolic boundary-layer partial differential equation and finally to an ordinary differential equation for film thickness.

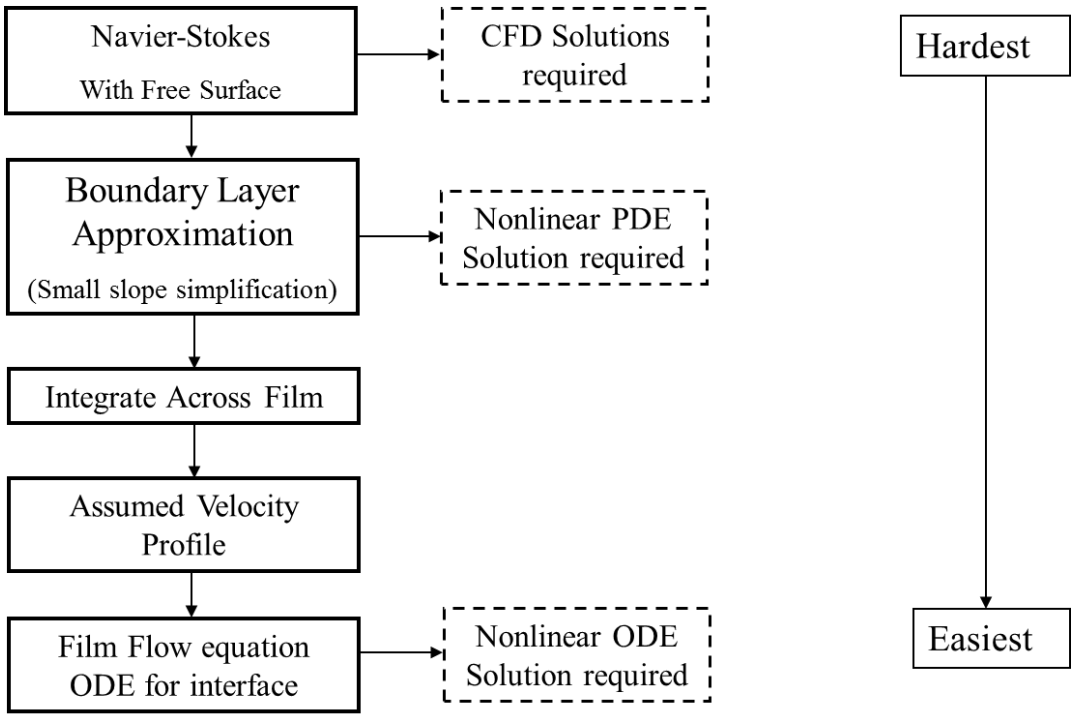

Figure 3.1: Dervation of film equation flow chart

Section 3.2 gives the derivation of the integral equation using the steps shown in the first three boxes of figure 3.1. This equation is used to create the different film equations with velocity profile assumptions which are the last two steps required to create a film

equation in figure 3.1. Section 3.3 and 3.4 derive two film equations using two different velocity profile assumptions. Section 3.3 gives the derivation of the parabola film equation using a parabolic velocity profile assumption in the integral equation. This is the most commonly used velocity profile assumption. Section 3.4 gives the derivation of the plug-parabola film equation using a plug velocity profile for the inertial terms. The traditional form of the film equation is the parabola film equation. The plug-parabola film equation is proposed in this research as a possible way to solve the deficiencies of the fully parabolic equation.

3.2 Boundary-Layer Approximation and Integral Equation Derivation

Consider the problem shown in figure 3.1 where the substrate is drawn from a liquid pool at a speed, S, with a constant film thickness, D, obtained downstream. The X and Y coordinate system is indicated in the figure and the interface is located at Y=H(X). The X and Y components of velocity are given as U and V respectively.

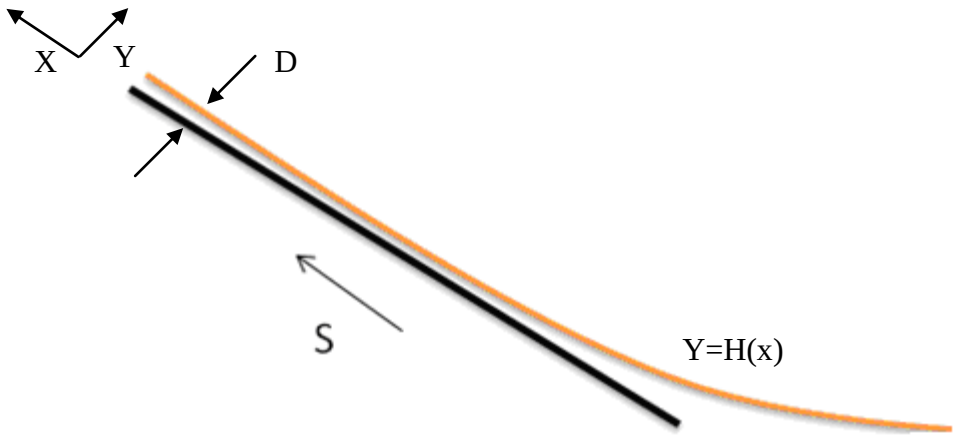

Figure 3.2: Set-up of boundary-layer approximation derivation

The boundary-layer approximation to the Navier-Stokes equation is made because the streamlines in thin film flows are nearly parallel .Variables are made dimensionless following the work of Landau and Levich who determined the asymptotic behavior of the film-forming flow at low speeds using the parameters [4]:

$$p = \frac{PD}{\mu S \delta} \quad x = \frac{X\delta}{D} \quad v = \frac{V}{\delta S} \quad u = \frac{U}{S} \quad y = \frac{Y}{D} \quad h = \frac{H}{D} \tag{3.1a}$$

where P is the dimensional pressure, μ is the dimensional viscosity and δ is a small parameter given below.

Upon substituting into the governing equations, the following dimensionless groups arise:

$$Ca = \frac{\mu S}{\sigma} = \delta^3 \qquad Bo = \frac{\rho D^2 G}{\mu S} \qquad \mathrm{Re} = \frac{\rho SD}{\mu} \qquad \xi = \frac{Q}{SD} \tag{3.1b}$$

Ca is the capillary number, Bo the Bond number, Re the Reynolds number, and ξ a dimensionless flow rate. Where the dimensional variables are; surface tension, σ; density ρ, G gravity; and the flow rate Q. The x and y components of the boundary-layer approximation to the Navier-Stokes equations are given below:

$$\mathrm{Re}\,\delta\left[u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right] = -\frac{\partial p}{\partial x} + \frac{\partial^2 u}{\partial y^2} - Bo\cos(\alpha) \tag{3.2a}$$

$$0 = -\frac{\partial p}{\partial y} + \delta Bo\cos(\alpha) \tag{3.2b}$$

Equation 3.2b arises because the viscous and inertial forces in the y-component of the Navier-Stokes equation are small compared with the pressure and the gravity terms. The continuity equation and the kinematic boundary equation conditions are unaffected in form by the small-slope approximation and, are given as:

$$0 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \tag{3.2c}$$

$$v = u\frac{dh}{dx} \quad \text{at} \quad y = h(x) \tag{3.2d}$$

The kinematic boundary condition makes the surface of the film a streamline. The normal stress boundary condition at the free surface is simplified for a small slope to give a relationship between the pressure and the dimensionless interfacial curvature, κ as:

$$p = -\kappa \qquad \kappa = \frac{\frac{d^2h}{dx^2}}{\left[1+\delta^2\left(\frac{dh}{dx}\right)^2\right]^{3/2}} \quad \text{at} \quad y = h(x) \tag{3.2e}$$

The dimensional curvature is κ divided by the final film thickness D. The pressure at the surface of the liquid is the product of surface tension and surface curvature. In order to describe a static meniscus once the dynamic flow terms have died out, the expression for curvature is not simplified. The shear free condition at the free surface is given in equation 3.2f. Another boundary condition is the no slip condition at the wall given in 3.2g.

$$0 = \frac{\partial u}{\partial y} \quad \text{at} \quad y = h \tag{3.2f}$$

$$u = 1,\ v = 0 \quad \text{at} \quad y = 0 \tag{3.2g}$$

Integrating equation 3.2b and applying the boundary condition 3.2e yields an equation for the pressure field.

$$p = \delta Bo\sin(\alpha)\left(y - \bar{h}\right) - \kappa \tag{3.2h}$$

We now develop the integral equation equivalent to the system of equations 3.2 above. The integral equation is obtained by taking the boundary-layer approximation of the x-

component of the Navier-Stokes equations (3.2a) and integrating across the film then inserting the continuity equation (3.2c), using Leibniz Rule, and applying the boundary conditions which gives the following result:

$$\xi = \int_0^h u dy \tag{3.3a}$$

$$\mathrm{Re}\,\delta \frac{d}{dx}\int_0^h u^2 dy = -\frac{\partial p}{\partial x} h - \frac{\partial u}{\partial y}_{y=0} - Bo\cos(\alpha)h \tag{3.3b}$$

Equation 3.3b is a convenient form of the momentum balance where no assumption has been made on the form of the velocity profile. A film equation can be obtained by inserting a form for the velocity profile in the system 3.3.

3.3 Parabola Film Equation Derivation

A parabolic velocity profile is assumed to satisfy the shear free condition at the air-liquid interface 3.2f, the no slip boundary condition 3.2g and the volumetric flow constraint 3.3a. After the constants have been solved, the resulting second order polynomial for the velocity profile is:

$$u = 1 + \frac{3(\xi - h)}{h^3}\left[hy - \frac{1}{2}y^2\right] \tag{3.4a}$$

The parabola film equation is obtained by substituting the velocity profile, 3.4a, into the integral equation, 3.3b, and solving. The resulting film equation is given below.

$$\mathrm{Re}\,\delta\left(\frac{1}{5h}\right)\frac{dh}{dx}\left[1 - \frac{6\xi^2}{h^2}\right] = \delta Bo\sin(\alpha)\frac{dh}{dx} + \frac{d\kappa}{dx} + \frac{3(\xi - \xi)}{h^3} - Bo\cos(\alpha) \tag{3.4b}$$

The parabola film equation is a good approximation for no inertia, but when inertia becomes important the parabola does not limit back to a hydrostatic meniscus. A hydrostatic meniscus is determined by surface tension and a hydrostatic pressure field. The main deficiency in the parabola film equation is that the inertial terms do not die out within the film equation as film thickness increases. As a result, a hydrostatic meniscus is not obtained far from the wall. This deficiency is explained in more detail in section 3.4 and shown in section 5.5.

3.4 Plug-Parabola Film Equation Derivation

For the purpose of evaluating momentum, it is assumed that there is plug flow within the film with no variation in the velocity profile within the y direction that satisfies 3.3a, the volumetric flow constraint. Using a plug flow for inertial terms is a common practice in pipe flow calculations. The plug-parabola flow equation is obtained by substituting the plug velocity profile 3.5a into the left-hand side of equation 3.3b. An equation for the shear along the wall is still obtained using the parabola velocity profile 3.4a. The velocity profile in equation 3.4a is substituted into the right hand side of equation 3.3b. The resulting equation is the proposed plug-parabola flow film equation given below:

$$u = \frac{\xi}{h} \tag{3.5a}$$

$$\mathrm{Re}\,\delta\left[-\frac{\xi^2}{h^3}\frac{dh}{dx}\right] = \delta Bo\sin(\alpha)\frac{dh}{dx} + \frac{d\kappa}{dx} + \frac{3(1-\xi)}{h^3} - Bo\cos(\alpha) \tag{3.5b}$$

Equations 3.5b and 3.4b differ in only the form of their left-hand sides, the inertia terms. The viscous term, third on the right, dies off as $\frac{1}{h^2}$. When the inertial terms in 3.4b and 3.5b are compared, it is seen that both equations have a term $\frac{1}{h^3}\frac{dh}{dx}$, which dies off, but only the parabola equation has the larger term, $\frac{1}{h}\frac{dh}{dx}$,which does not die off far from the

wall. When the Reynolds number is zero, no inertia, equations 3.4b and 3.5b are the same. The plug-parabola film equation was proposed because it shows the desired ability to limit back to a hydrostatic meniscus. Precedent for evaluating the inertial term using a plug velocity profile is found in pipe flow and jet problems where momentum balances are used. The viscous term is evaluated as a friction factor which is often experientially determined.

Chapter 4
Film Equation Velocity Profile Search

4.1 Introduction

The integral equation, derived in Chapter 3 and shown again below for convenience, requires the assumption of a velocity profile in order to be solved.

$$\mathrm{Re}\,\delta \frac{d}{dx}\int_0^h u^2 dy = -\frac{\partial p}{\partial x}h - \left.\frac{\partial u}{\partial y}\right|_{y=0} - Bo\cos(\alpha)h \qquad (4.1)$$

Once this velocity profile is assumed and the integral evaluated, the result is a film equation expressed solely in terms of the interface location h. The most commonly chosen velocity profile is a parabola; however, the resulting film equation breaks down when inertial effects become important and the film equation cannot limit back to a static meniscus. The starting hypothesis of this research is that the effectiveness of a film equation is related to the velocity profile chosen. In order to achieve a better film equation, a better velocity profile must be assumed. This chapter details the search for a velocity profile by observing what affect inertia has on the velocity profile in three different numerical solutions: a wedge problem, a slot die problem and a solution to the full boundary-layer equation that does not assume a velocity profile.

4.2 Wedge Problem

Problem Setup and Definition

Flow in a wedge with one wall moving and one wall shear free was solved to represent a small section of the meniscus shape near the pool using Fluent version 6.3.26. The inlet velocity profile was fully developed at the entrance to this section of the meniscus shape with the desired flow rate. The flow field was examined for any significant deviations

from the parabolic velocity profile. Another reason for solving the wedge problem in Fluent was to gain experience solving fixed boundary problems.

Figure 4.1 shows the layout and boundary conditions and geometry of the wedge problem. Dimensional parameters were chosen for the geometry because Fluent requires dimensional inputs in order to run. The entrance height H was chosen to be 5cm, an exit height h of 1cm, 20% of the inlet height, the wedge angle α was chosen to be 15 degrees. The angle α was chosen to be substantial enough to allow for easy meshing but not totally outside the small slope approximation. The length L was determined by trigonometry.

Within Fluent, the boundary conditions were set as shown in figure 4.1. The bottom wall was set to be a moving wall in the negative x direction. The inlet velocity profile used for the wedge problem is derived in Appendix A-2. The solver set-up for Fluent as well as the mesh generation within Gambit is detailed in Appendix A.

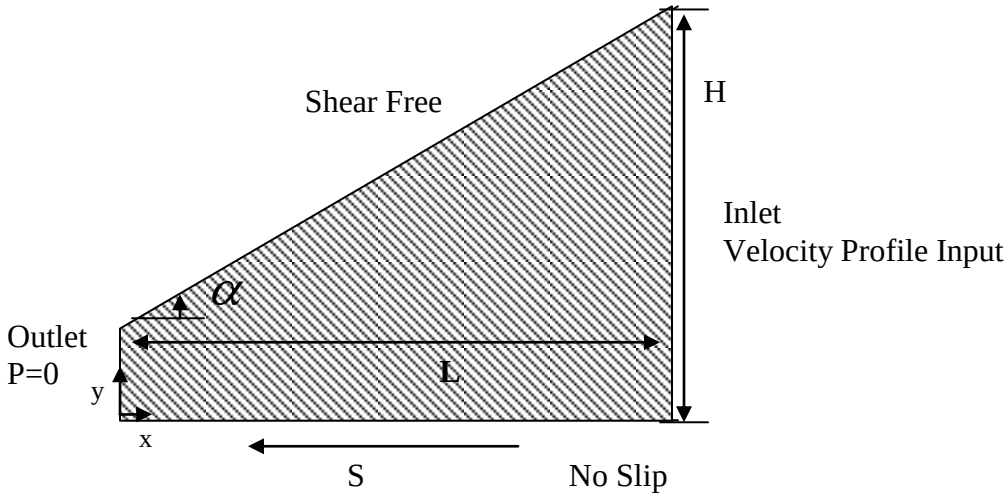

Figure 4.1: Wedge problem boundary conditions and geometry

Wedge Data and Plots

The graphs shown below were created with a density of 1 (kg/m^3), alpha of 15 degrees and a wall speed of 200 (m/s). The values for viscosity were found using equation 4.2 found from the dimensionless groups of Reynolds number and flow rate λ. Fluent requires dimensional input parameters and the wedge problem was solved dimensionless. Therefore, the values used to solve the wedge problem in Fluent were chosen because they gave rise to the correct values for the dimensionless groups.

$$\mu = S\rho H\left(\frac{\lambda}{\mathrm{Re}}\right) \tag{4.2}$$

where $\lambda = \frac{q}{SH}$ and $\mathrm{Re} = \frac{\rho SH}{\mu}$

Three Reynolds numbers were chosen and run at three different flow rates. The Reynolds numbers chosen were 1, 20, and 50. These Reynolds numbers were chosen because they span the region where there is little to high importance of inertia respectively. The flow rates were 0.01, 0.02, and 0 to reflect that in dip coating there is reverse flow. A small value for the flow rate corresponds to a large value of h in a film flow (static meniscus area), while a larger value of the flow rate corresponds to a smaller value of h nearer the final film thickness. The velocity profile was analyzed at four different locations for each set of data: the inlet, half way from the outlet, three quarters from the outlet and seven eighths from the outlet.

The plots shown below are dimensionless such that the wall velocity is 1 and distance between walls is 1. At Reynolds number 1, the velocity profile for each flow rate at each location within the wedge is a parabola as expected for viscous flow shown in figure 4.2. This shape was determined by curve fitting a parabola to the data.

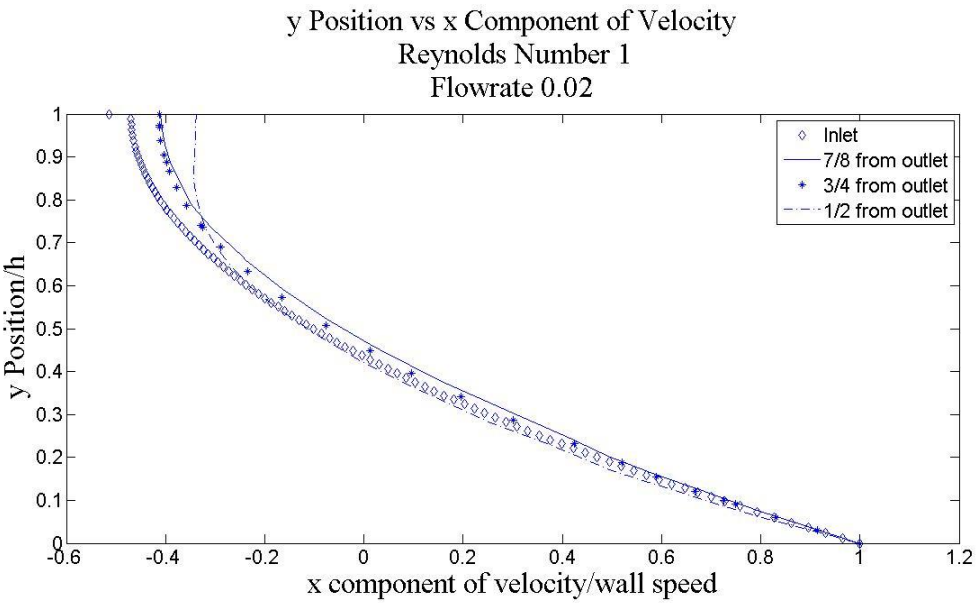

Figure 4.2: y position verses x component of velocity with a Reynolds number of 1 and a flow rate of 0.02

Figures 4.3 and 4.4, for Reynolds numbers of 20 and 50 at a flow rate of 0.02, show the formation of boundary layers along the shear free and no slip boundaries. A center portion of the velocity profile appears to be linear. A boundary layer refers to a change in profile shape near a wall to accommodate the boundary condition at that wall.

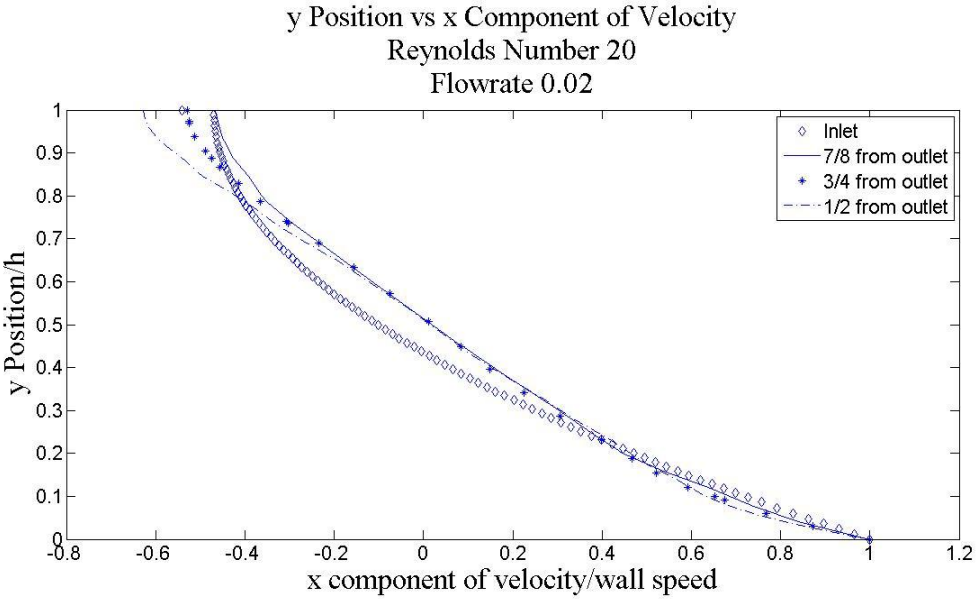

Figure 4.3: y position verses x component of velocity with a Reynolds number of 20 and a flow rate of 0.02

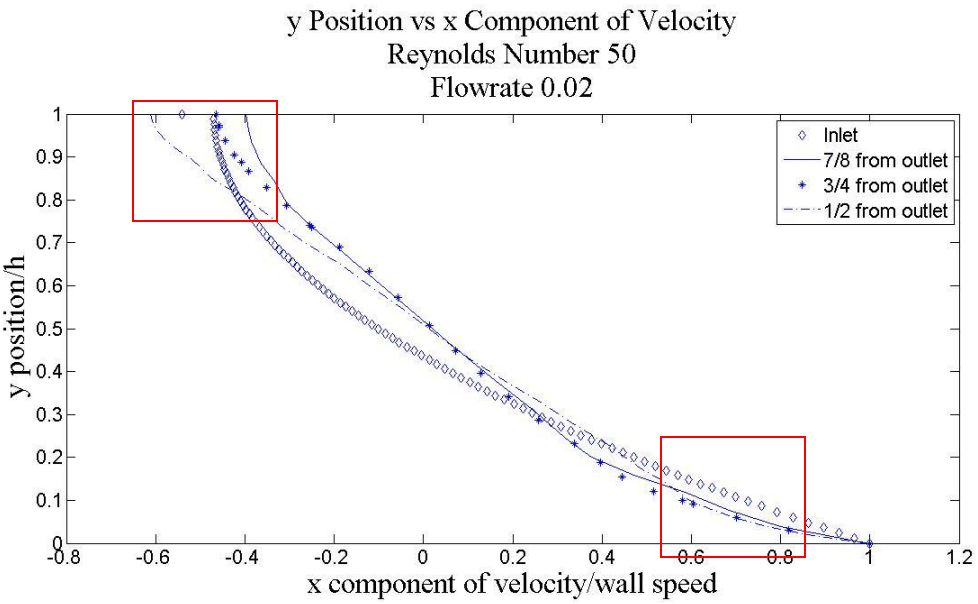

Figure 4.4: y position verses x component of velocity with a Reynolds number of 50 and a flow rate of 0.02

On the other hand, boundary layers are not noticeable at the larger flow rate of 0.2 representing flow nearer the final film thickness. A near parabola velocity profile is seen in Figure 4.5 for a Reynolds number of 50 and a flow rate of 0.2. A parabola was fitted to the velocity profile seven eights from the inlet and is shown in figure 4.6.

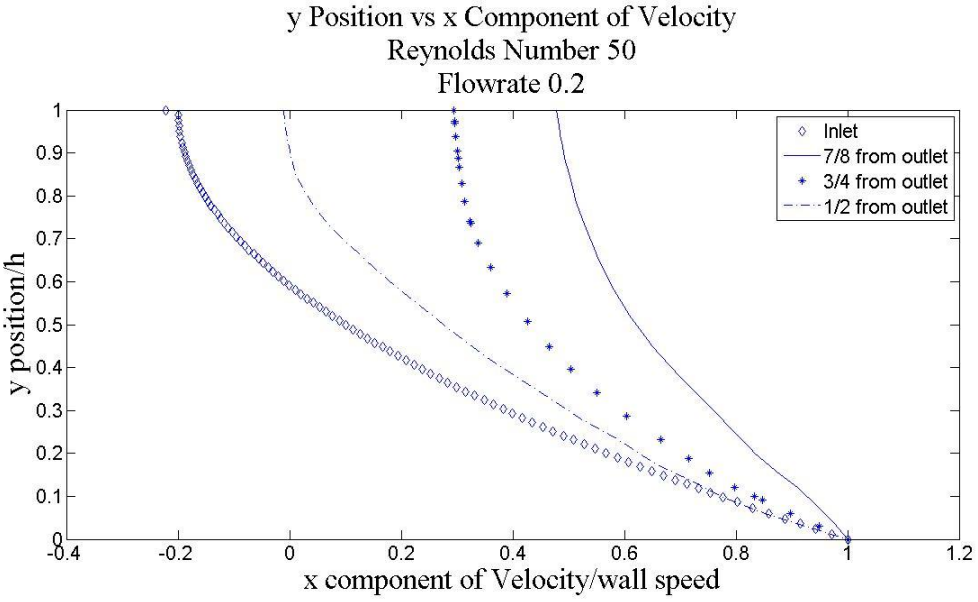

Figure 4.5: y position verses x component of velocity with a Reynolds number of 50 and a flow rate of 0.2

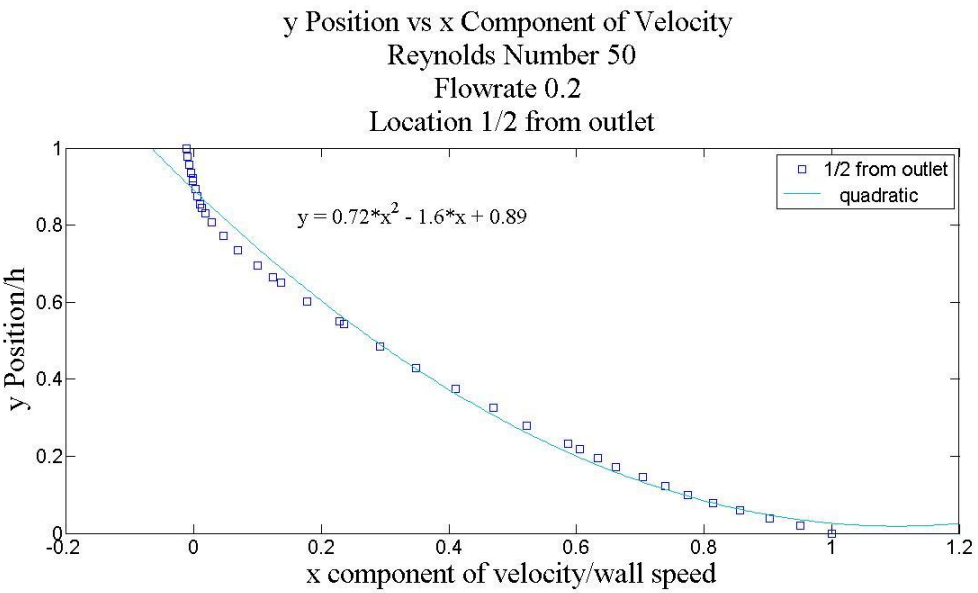

Figure 4.6: y position verses x component of velocity at 1/2 from outlet with a Reynolds number of 50 and a flow rate of 0.2 with a fitted parabola

Summary

The results for flow in a wedge suggest that boundary layers develop for small dimensionless flow rates (0.02) corresponding to large values of h, that is in the region of the static meniscus where the flow terms should have little importance. The profile is still grossly parabolic and the departure is not likely important in the static meniscus region. For larger flow rates (0.2) representing a few film thicknesses where hydrodynamic effects are more important, the velocity profile more closely resembles a parabola. From the wedge solution plots there is no clear case for abandoning a parabolic profile assumption in the creation of a film equation. However, a single profile that can involve two boundary layers, shown in figure 4.4, is unlikely. If the boundary layers are critical, then the quest for an improved film equation is almost certainly lost.

4.3 Slot Die Coating Problem

Problem Setup and Definition

The original intention was to try the free surface dip coating problem in Fluent. However, Fluent uses the volume of fluid method where part of the cell represents the liquid and the other part the air. Such a method is unlikely to produce an accurate meniscus shape. Instead, a more realistic film profile was generated and used as a fixed boundary for Fluent to avoid the free surface flow. Kam Ng, a mathematician from Kodak, was asked to try the free surface problem using Flow 3D which also uses the volume of fluid method. His attempts produced clearly unrealistic artifacts, and so the decision to avoid the volume of fluid method was supported.

A 2-D slot die coater as shown in Fig. 4.7 was modeled and solved using Fluent 6.3.26. In reality, the changes in the interface shape and velocity profile are coupled. To simplify the numerical algorithm and to remain within the capabilities afforded by Fluent, a reasonable interface shape, found from the parabola film equation (3.4b), was imported for the velocity profile evaluation. Subsequently, the use of the parabola film equation for the interface shape was proven to be an accurate approach. The parabola film equation's ability to describe the interface shape accurately is shown in Chapter 5.

Figure 4.7 shows the layout of the slot die problem. The entrance height W is ten times the final film thickness D. This was picked because it would produce thick films and would not cause meshing problems. However, very thick films as considered in the wedge flow calculation are not represented. The boundary condition on the outlet is pressure equals zero. The inlet boundary condition is a fully developed velocity profile derived in Appendix B. The bottom wall has a no-slip boundary condition as well as the top wall. The meniscus shape/curve is a shear free boundary condition. The bottom wall in the problem has a selected wall speed in the positive x direction.

An outlet boundary condition was used within Fluent for the outlet. For more detailed information about the mesh generation in Gambit as well as detailed solver set-up within Fluent see Appendix B.

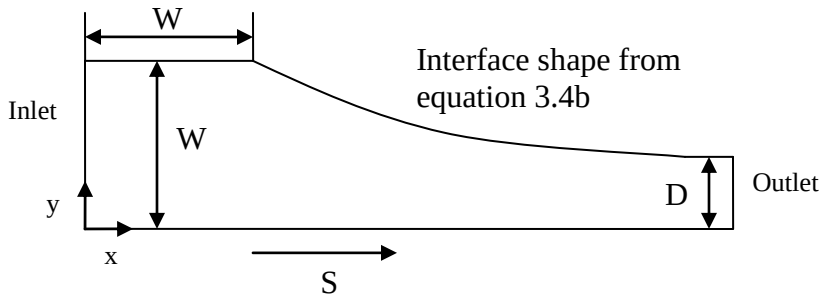

Figure 4.7: Slot die model configuration

Slot Die Data and Plots

As shown in Appendix B, Fluent was coded to accept dimensional values; however, dimensionless values were needed for our model. The equations used to find the viscosity and wall speed are shown in equations 4.4 and 4.5 respectively. In order to solve for the required dimensional inputs for Fluent, wall speed and viscosity, three dimensionless group values had to be specified. These values were the capillary number, Reynolds number and a dimensionless flow rate of 1. Because of the scaling used, the dimensionless flow rate is one when gravity is neglected.

The graphs shown below were created with a density of 1 (g/m^3), a final film thickness of 1 mm and surface tension 1 (g/s^2). Equations 4.4 and 4.5 were found using equation set 4.3. The values for viscosity and wall speed were found using the dimensionless groups in equation 4.4 and 4.5 respectively.

$$Ca = \delta^3 = \frac{\mu S}{\sigma} \tag{4.3a}$$

$$\mathrm{Re} = \frac{\rho SD}{\mu} \tag{4.3b}$$

$$q = SD \tag{4.3c}$$

$$\mu = \sqrt{\rho D \sigma \left(\frac{Ca}{\mathrm{Re}} \right)} \tag{4.4}$$

$$S = \mathrm{Re}\left(\frac{\mu}{\rho D} \right) \tag{4.5}$$

Fluent runs were completed for four Reynolds numbers and two capillary numbers. The Reynolds numbers were 0.5, 10, 25, 50 and the capillary numbers were 0.1 and 0.01. The plots shown below are created using dimensionless variables where speed at the wall is 1 and distance across the film is 1. Figures 4.9, 4.11, 4.13 and 4.15 show the meniscus shape from the parabola film equation used to mesh the slot die problem in Fluent at each Reynolds number and capillary number.

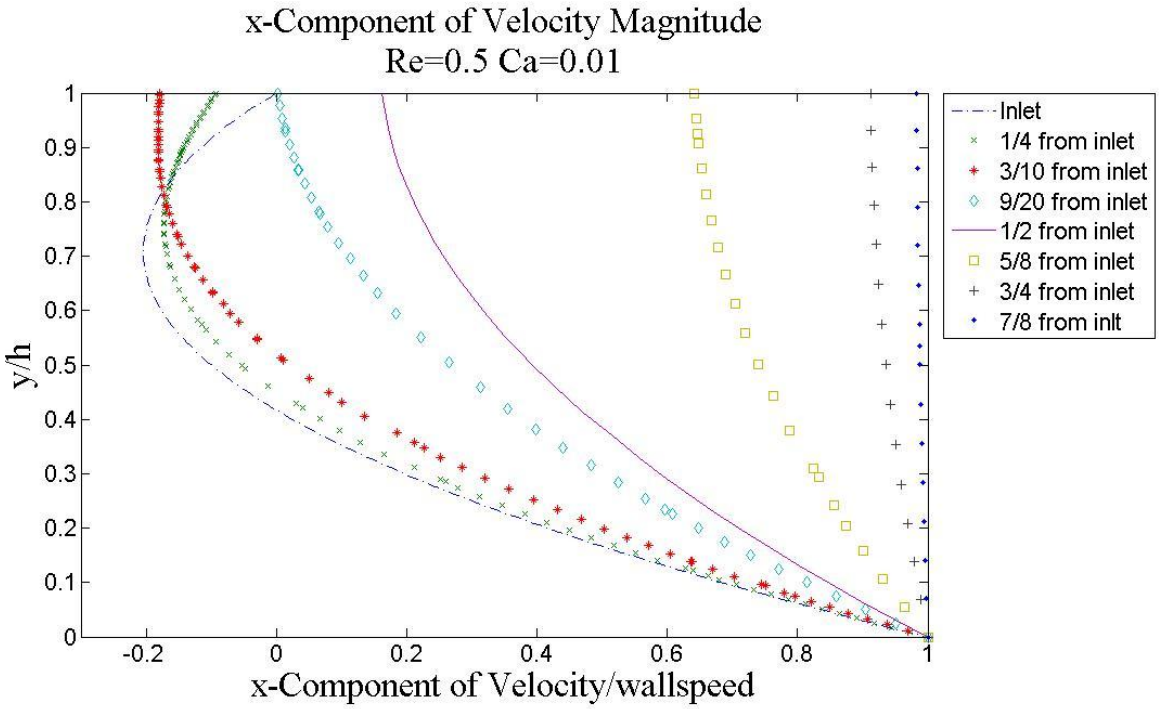

Figure 4.8: y position verses x component of velocity with a Reynolds number of 0.5 and a capillary number of 0.01

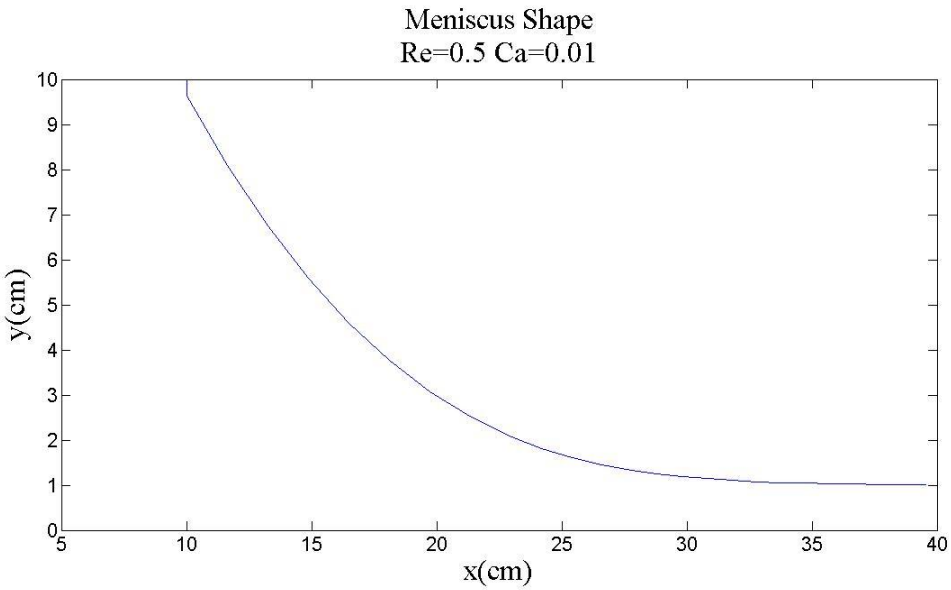

Figure 4.9: Meniscus shape Reynolds number of 0.5 and a capillary number 0.01

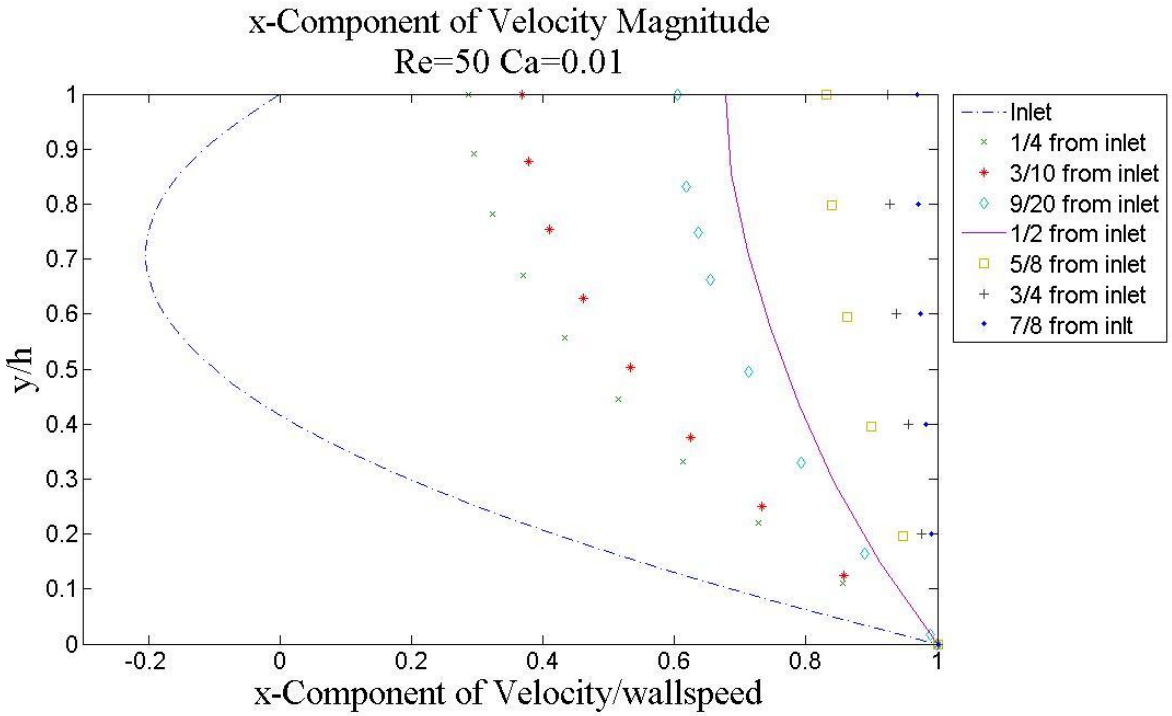

Figure 4.10: y position verses x component of velocity with a Reynolds number of 50 and a capillary number of 0.01

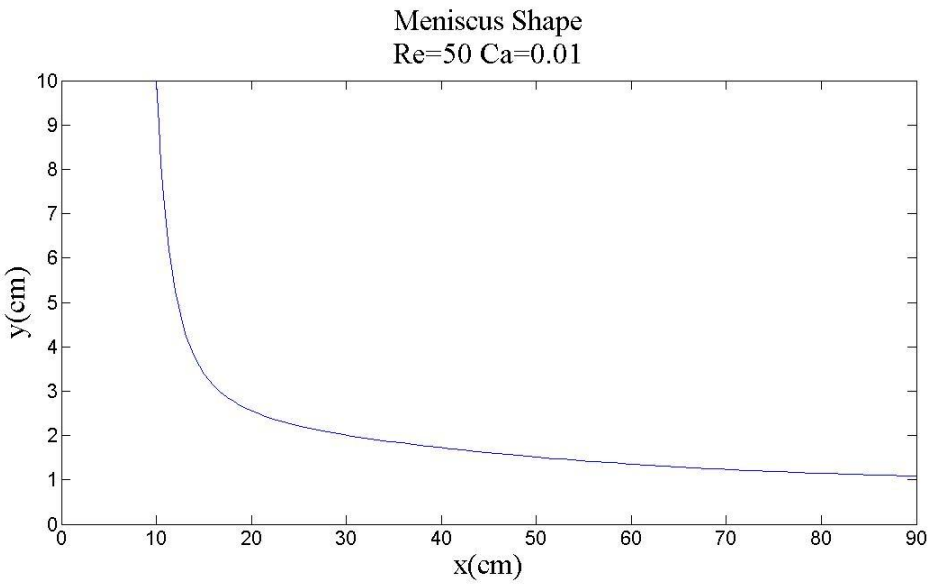

Figure 4.11: Meniscus shape Reynolds number of 50 and a capillary number 0.01

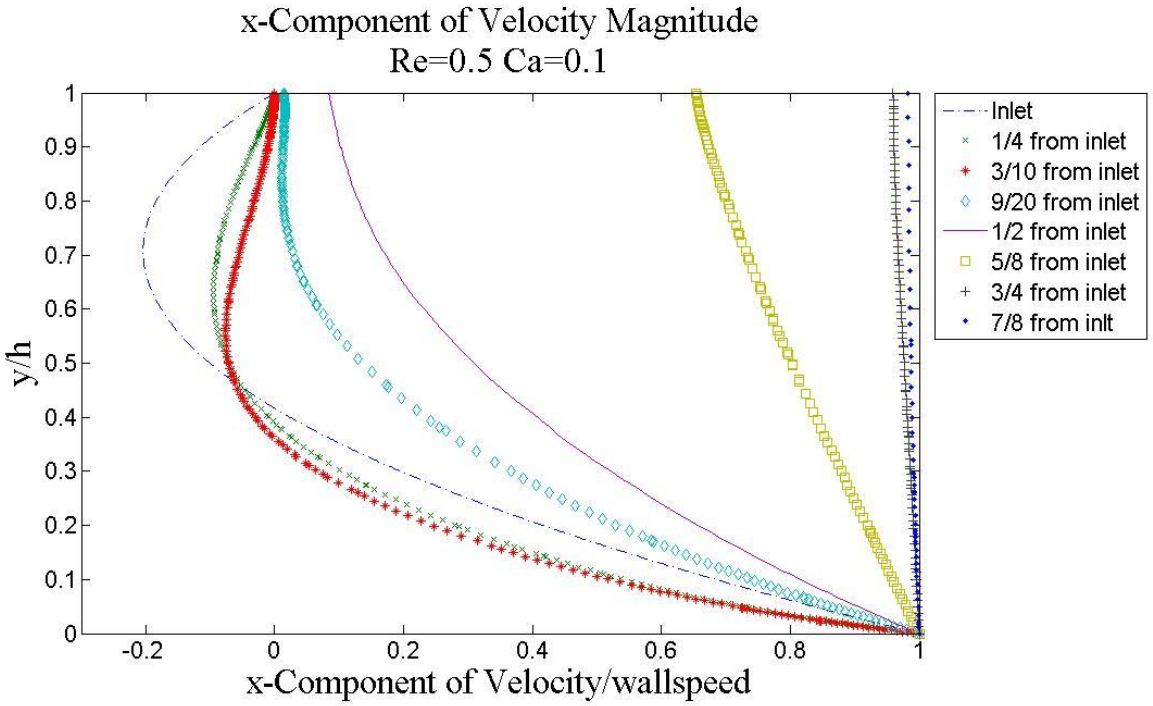

Figure 4.12: y position verses x component of velocity with a Reynolds number of 0.5 and a capillary number of 0.1

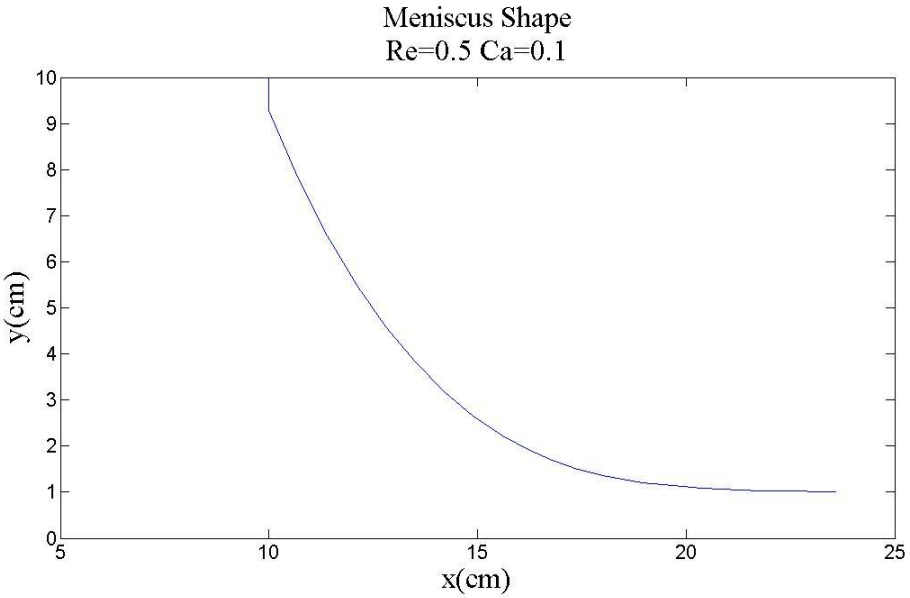

Figure 4.13: Meniscus shape Reynolds number of 0.5 and a capillary number 0.1

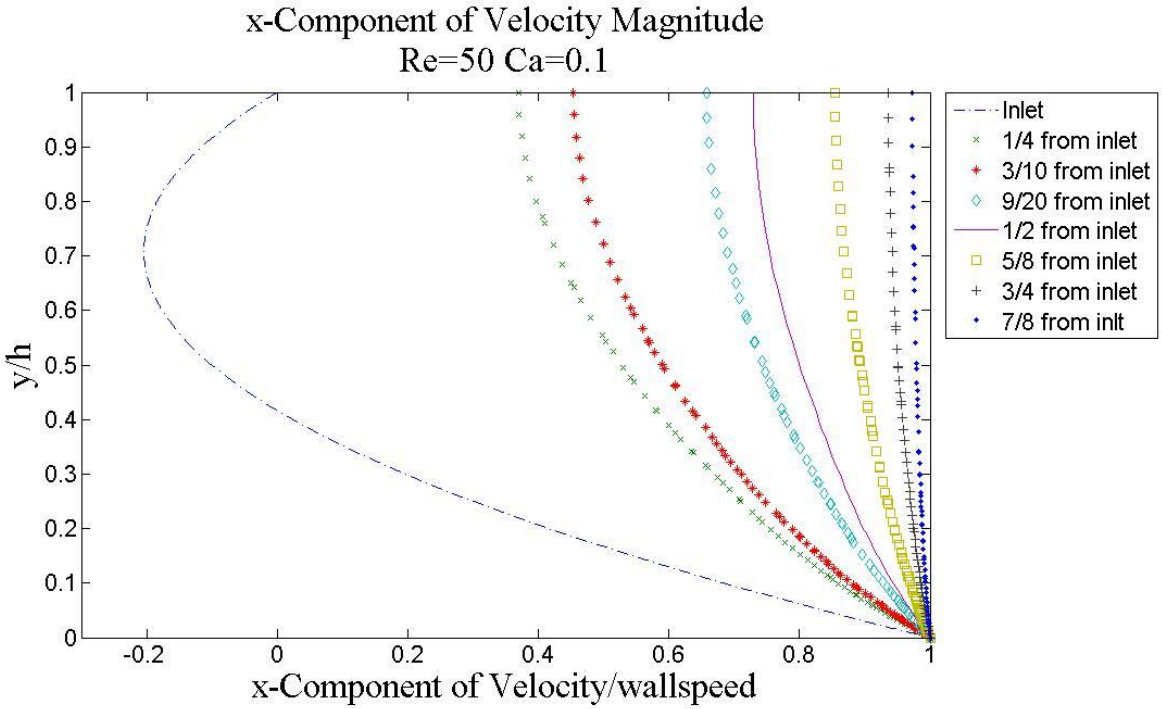

Figure 4.14: y position verses x component of velocity with a Reynolds number of 50 and a capillary number of 0.1

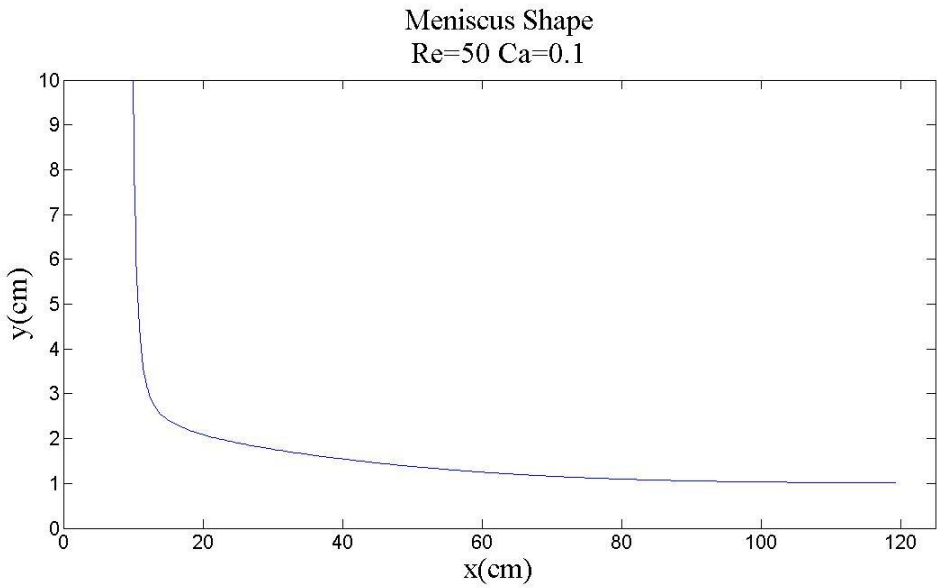

Figure 4.15: Meniscus shape Reynolds number of 50 and a capillary number 0.1

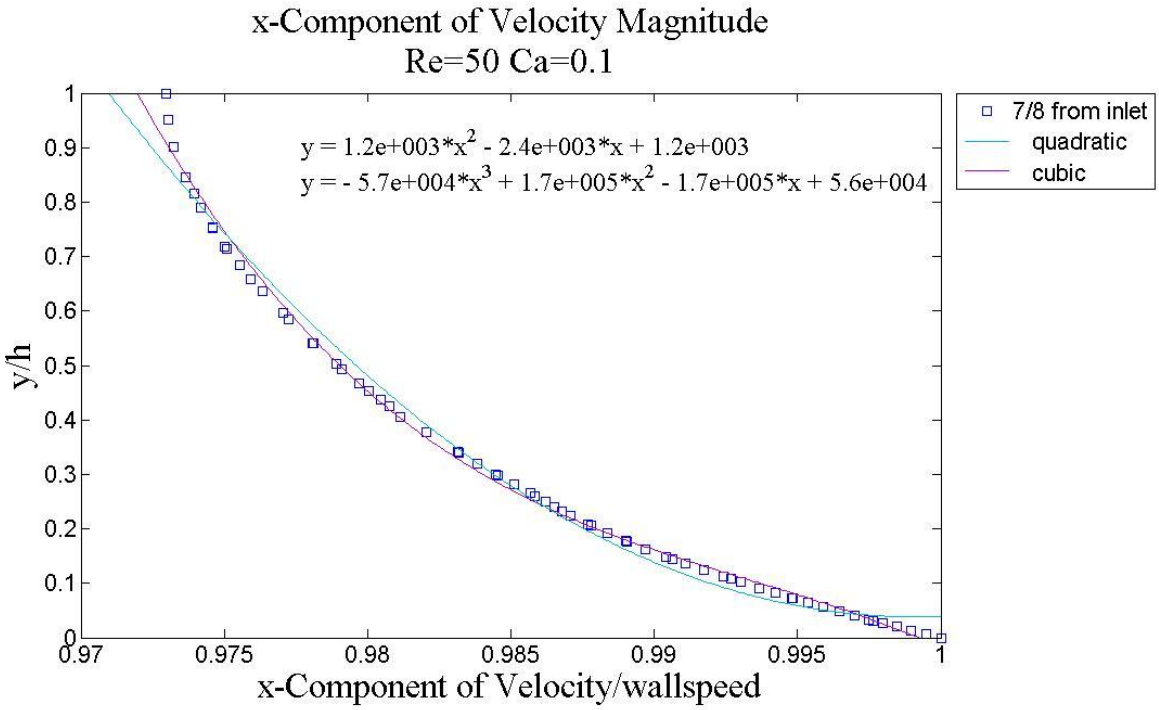

Figure 4.16: y position verses x component of velocity with a Reynolds number of 50 and a capillary number of 0.1 fitted quadric and cubic

From the slot die data, it was seen that the velocity profiles throughout the domain were similar in shape to parabolas even when the Reynolds number was large. This is seen in figures 4.8, 4.10, 4.12 and 4.14. This was particularly true away from the inlet, but even near the inlet no profile was grossly non-parabolic. It was also found that a cubic velocity

profile was slightly better at fitting the simulated velocity profiles for all Reynolds numbers, figure 4.16. The slot die showed no development of boundary layers at the shear free and at the no slip boundary walls as seen in the wedge problem, figure 4.4. The wedge problem modeled thicker films than the slot die problem explained in section 4.2. This supports the contention made in section 4.2 that the boundary layers seen in the wedge problem are confined to large values of film thickness. When there are large values of film thickness hydrodynamic effects are less important therefore the boundary layers can be ignored. The main conclusion of the model problems done with Fluent is that the assumption of a parabolic shape for the velocity profile is reasonable.

Summary

Both the wedge and slot die problems support the assumption of a parabolic velocity profile for deriving the film flow equation. Therefore, the starting hypothesis that the velocity profile was the blame for the deficiency of the film equation was shown to be incorrect. It was this finding that lead to the proposed plug-parabola film equation.

4.4 Linear Full Boundary-Layer Approximation

The full boundary-layer approximation (full PDE) was solved, Appendix D, to help understand the velocity profile within the film as well as to see how the small slope approximation affects the velocity profile. A possible velocity profile, when gravity is neglected, for the film equation was found from the linear solution of the full PDE. The terms substituted into equation D.5 from Appendix D are shown below, 4.7a. Equation D.5a from Appendix D is shown below for convenience, equation 4.6. $\overline{F}$ is the stream function for the parabolic flow far downstream. The equation is linearized about the flow far downstream.

(4.6)

$$\mathrm{Re}\,\delta\left[\left(-\frac{1}{h^3}\right)\frac{dh}{dx}\left(\frac{\partial F}{\partial \eta}\right)^2+\left(\frac{1}{h}\right)\frac{\partial^2 F}{\partial x\partial y}\frac{\partial F}{\partial \eta}-\left(\frac{1}{h^2}\right)\frac{\partial^2 F}{\partial \eta^2}\frac{\partial F}{\partial x}\right]=\frac{\partial \kappa}{\partial x}+\left(\frac{1}{h^3}\right)\frac{\partial^3 F}{\partial \eta^3}+\delta Bo\sin(\alpha)\frac{dh}{dx}-Bo\cos(\alpha)$$

where $\eta=\frac{y}{h}$

$$h=1+h' \qquad F=\overline{F}+F' \tag{4.7a}$$

The value of $\overline{F}$ is found in Appendix D equation D.5f shown below for reference.

$$\overline{F}=Bo\left[\frac{\eta^3}{6}-\frac{\eta^2}{2}\right]\cos(\alpha)+\eta \quad x\to\infty \tag{4.7b}$$

To linearize the problem, the partial derivatives of 4.7a are found and equation 4.6 is put in terms of $\overline{F}, F' \text{ and } h'$.

$$\mathrm{Re}\,\delta\left[\frac{\partial^2 F'}{\partial x\partial \eta}\left(\frac{\partial \overline{F}}{\partial \eta}\right)-\left(\frac{\partial \overline{F}}{\partial \eta}\right)^2\frac{dh'}{dx}-\frac{\partial^2 \overline{F}}{\partial \eta^2}\frac{\partial F'}{\partial x}\right]=\frac{d^3h'}{dx^3}+\delta Bo\sin(\alpha)\frac{dh'}{dx}+\frac{\partial^3 F'}{\partial \eta^3}-3h'Bo\cos(\alpha) \tag{4.8a}$$

The boundary conditions in terms of $F' \text{ and } h'$ are listed below:

$$\frac{\partial F'}{\partial \eta}=h' \quad \text{at} \quad \eta=0 \tag{4.8b}$$

$$F'=0 \quad \text{at} \quad \eta=0 \tag{4.8c}$$

$$\frac{\partial^2 F'}{\partial \eta^2}=0 \quad \text{at} \quad \eta=1 \tag{4.8d}$$

$$F' = 0 \qquad \text{at} \quad \eta = 1 \tag{4.8e}$$

The equation set 4.8 is solved by recognizing that the solution to the problem is separable and exponential in the x direction. The constant A, a small maximum amplitude, and Φ is a function of η only. The constant m gives the rate of decay of the solution in the x direction and its value is determined as part of the solution.

$$h' = Ae^{-mx} \qquad F' = A\Phi e^{-mx} \tag{4.9}$$

The derivatives of the exponentials, equation 4.9, are found and substituted into equation set 4.8. The result is shown below in equation set 4.10.

$$\mathrm{Re}\,\delta m\left[\frac{\partial\Phi}{\partial\eta}\left(\frac{\partial\overline{F}}{\partial\eta}\right)-\left(\frac{\partial^2\overline{F}}{\partial\eta^2}\right)\Phi\right]+\frac{\partial^3\Phi}{\partial\eta^3}=m^3+mBo\,\delta\sin(\alpha)+3Bo\cos(\alpha)+\mathrm{Re}\,m\delta\left(\frac{\partial\overline{F}}{\partial\eta}\right)^2 \tag{4.10a}$$

The boundary conditions in terms of Φ and η are listed below.

$$\frac{\partial\Phi}{\partial\eta}=1 \qquad \text{at} \quad \eta = 0 \tag{4.10b}$$

$$\Phi = 0 \qquad \text{at} \quad \eta = 0 \tag{4.10c}$$

$$\frac{\partial^2\Phi}{\partial\eta^2}=0 \quad \text{at} \quad \eta = 1 \tag{4.10d}$$

$$\Phi = 0 \qquad \text{at} \quad \eta = 1 \tag{4.10e}$$

The linear third order equation set 4.10 was solved in MatLab by finite differences. The unknowns are the values of Φ at the nodes and the rate of decay constant m. Normally a

third order problem has three boundary conditions; the fourth boundary condition is needed because of the additional unknown m.

Two special cases of equation set 4.10 can be solved analytically. When the Reynolds number is zero and there are no inertial terms, the differential equation for Φ is readily integrated to obtain a third degree polynomial. This solution is, in fact, the parabola film equation already obtained by assuming a parabolic velocity profile. The important conclusion is that in the absence of the inertial terms, the parabola film equation is the exact solution to the problem. Therefore, any film equation should reduce to the parabola film equation when the inertial terms are dropped.

The second special case where an analytic solution can be obtained is for no gravity. In that case, the limiting velocity profile far downstream is plug, not parabolic. When gravity is neglected, Bo=0, equation 4.6b reduces to equation 4.11 and the $\overline{F}$ partial derivatives are show below.

$$\overline{F} = \eta \qquad \frac{\partial \overline{F}}{\partial \eta} = 1 \qquad \frac{\partial^2 \overline{F}}{\partial \eta^2} = 0 \tag{4.11}$$

Equation 4.11 substituted into equation 4.10a gives a new linear equation, 4.12, without gravity. The constant β is used to simplify the integration of equation 4.12.

$$\beta^2 \left[\frac{d\Phi}{d\eta} \right] + \frac{d^3\Phi}{d\eta^3} = m^3 + \beta^2 \qquad \text{where} \qquad \beta^2 = \mathrm{Re}\, m\delta \tag{4.12}$$

Equation 4.12 is integrated and the integration constant is solved for using the boundary conditions 4.10d and 4.10e.

$$\beta^2 \Phi + \frac{d^2\Phi}{d\eta^2} = \left(\eta^3 + \beta^2 \right) \left(\eta - 1 \right) \tag{4.13}$$

The differential equation above, equation 4.13, is solved by finding the homogenous solutions, as well as the particular solution. The particular solution is found assuming a first order polynomial, Φ_p. The final solution Φ is given below.

$$\Phi = \frac{m^3 + \beta^2}{\beta^2}\left[\frac{\sin(\beta(-\eta))}{\sin(\beta)} - (-\eta)\right] \tag{4.14}$$

The values of m in equation 4.14 are found by applying boundary condition 4.10b. A cubic polynomial result was found for each m equation and m is the real root which was determined using Newton's method.

The stream function Φ, equation 4.14, needs to be converted back to the x component of velocity. The equations 4.9, 4.7a and D.4 from Appendix D are used to convert equation 4.14 to the x component of velocity, u.

$$u = \left(1 + \left\{\frac{m^3 + \beta^2}{\beta^2}\left[\frac{-\beta\cos(\beta(-\eta))}{\sin(\beta)} + 1\right]\right\}e^{-mx}\right)\left(\frac{1}{1 + e^{-.mx}}\right) \tag{4.15}$$

4.5 Linear Solution Comparisons

The plug-parabola film equation, parabola film equation and the full PDE were linearized as shown in section 4.4 for the full boundary-layer approximation solution. The linearized form of the parabola film equation and the plug-parabola film equation are shown in equations 4.16 and 4.17 respectively where We is the Weber number shown in equation 4.18.

$$\frac{We}{5}\left[-6\xi^2\frac{dh'}{dx} = \frac{d^3h'}{dx^3} + \delta Bo\sin(\alpha)\frac{dh'}{dx} + 3(-Bo\cos(\alpha)h'\right] \tag{4.16}$$

$$-We\left[\frac{dh'}{dx}=\frac{d^3h'}{dx^3}+\delta Bo\sin(\alpha)\frac{dh'}{dx}+3\left(-Bo\cos(\alpha)\right)h'\right] \tag{4.17}$$

$$We=\delta\,\mathrm{Re} \tag{4.18}$$

The linearized equations were solved by assuming an exponential solution for h' equation 4.19 has one fewer exponential solution than equation 4.9. Again m is the decay rate to be determined and A is a small maximum amplitude.

$$h'=Ae^{-mx} \tag{4.19}$$

Once equation 4.19 had been substituted into the linearized equations, a cubic polynomial in terms of m results where m is the real root. Newton's method was used to find the value of m at a given set of parameters.

4.6 Asymptotic Solution Comparisons

Figure 4.17 was created with the Bond number equal to zero, no gravity, so that ξ=1. This figure shows that for the case of no gravity, the value of m is the same for the parabola and plug-parabola film equations. Additionally, at small Re, this value is the same as the exact value from the full PDE. As We increases, the value of m from the film equations remains close to the exact result. Additional comparisons for non-zero Bo show similarly close values. The conclusion is that the asymptotic behaviors of the film equations are close to the exact result.

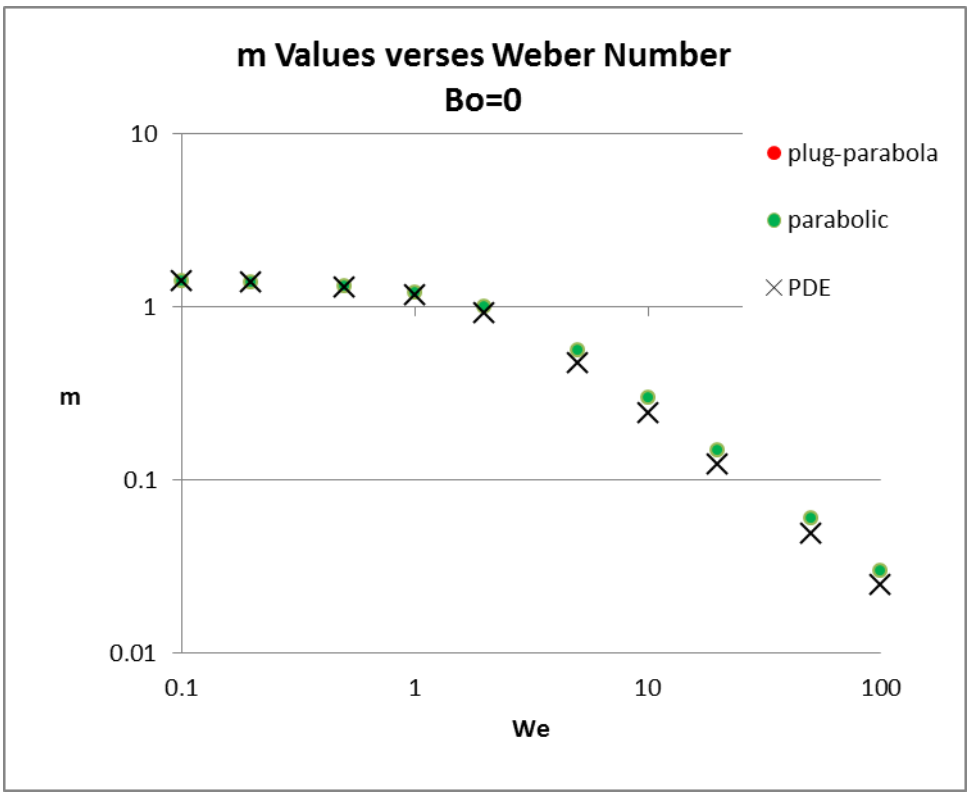

Figure 4.17: m values verses Weber number, Bo=0

The asymptotic velocity, $x \to \infty$, was found for the plug-parabola film equation, parabola film equation and the full boundary-layer approximation. The general velocity profile from the PDE is not parabolic when inertial effects are important. However, Figure 4.18 shows that the asymptotic velocities are very similar when bond number is zero with a high Weber number of 20. The conclusion is that a parabolic velocity profile gives a good approximation to the exact velocity profile even when inertial effects are large.

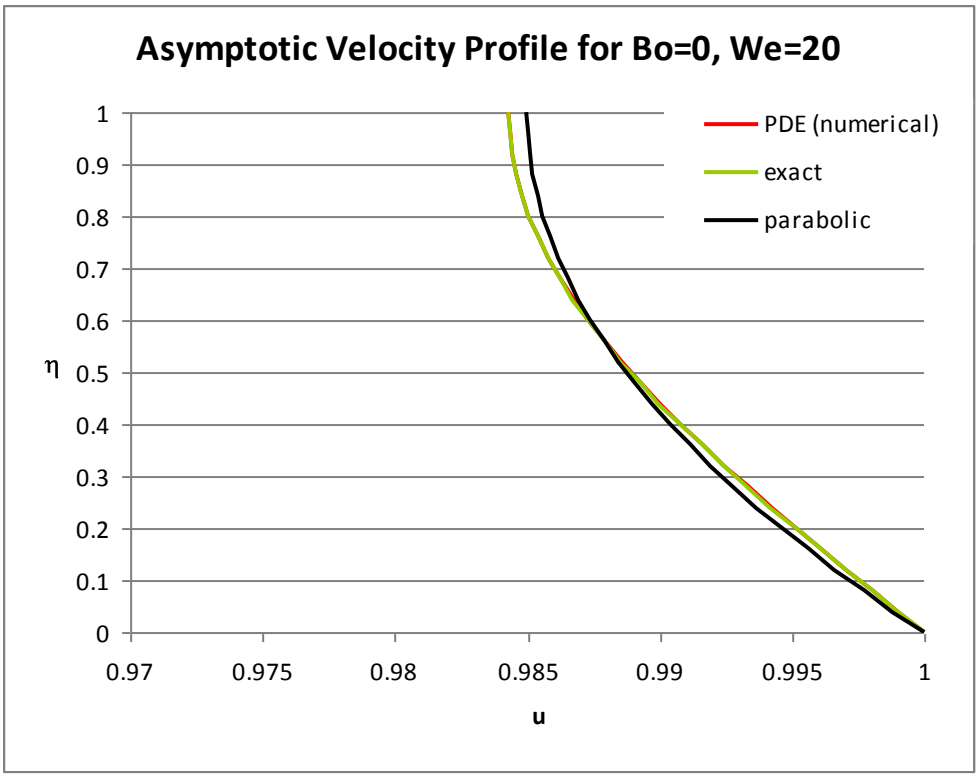

Figure 4.18: Asymptotic velocity profile for Bo=0 and We=20

Summary

The velocity profile shown in equation 4.15 can be fitted extremely well with a parabolic or cubic velocity profile. This is consistent with results from the slot die problem from section 4.3. Our overall conclusion is that an assumed parabolic or cubic velocity profile should be adequate to develop a film equation. Another important conclusion is that when Re=0, no inertia, the exact solution to equation 4.6, the (non-linearized) PDE, is the exact parabola from the parabola film equation represented already.

Chapter 5
Literature Comparison

5.1 Introduction

To test the plug-parabola film equation developed in Chapter 3, it was compared to results from Acrivos, Groenveld, Carvalho, and Dennis Coyle [2, 6, 7]. Dennis Coyle's data was obtained through private communication. Data was also obtained for the free surface runs in Flow 3D from Kam Ng through private communication. This data is a mix of experimental data, numerical calculations with FIDAP and private mathematical codes. The data found for dip coating does not extend into the regime of inertia making it difficult to test the plug-parabola film equation in a dip coating problem. As a result slot coater data was found and used to compare with the results from the parabola and plug-parabola film equations.

5.2 Dip Coating Meniscus Shape Comparison

Figure 5.1 shows that there is good agreement between the plug-parabola film equation and Acrivos's dip coating solution from FIDAP; however, Acrivos's paper does not have solutions for the region where inertial affects become important because convergence was lost and could not be used for this comparison [2]. When Re=0, no inertia, figure 5.1 shows good comparison even with a high capillary number. The plug-parabola film equation captures the whole meniscus back into the pool with all of the terms included in curvature equation after the small slope approximation.

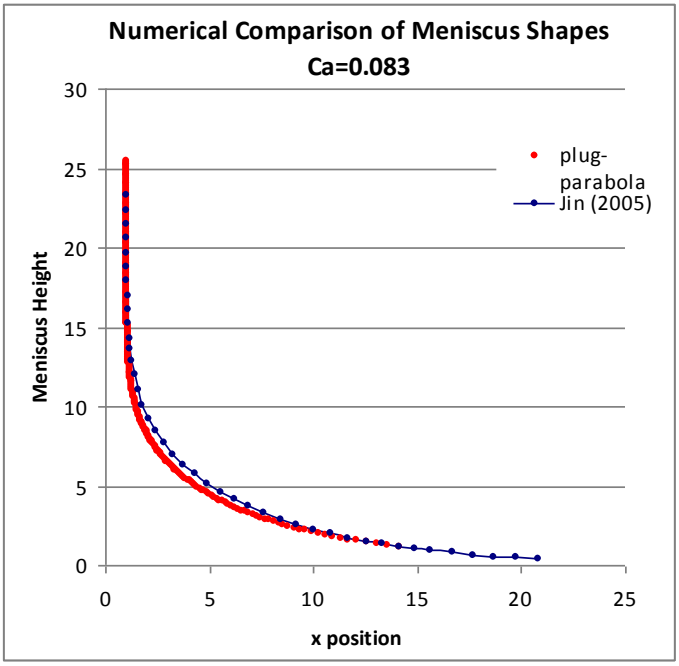

Figure 5.1: Comparison of plug-parabola versus Acrivos literature data

However, Fig. 5 from Jin and Acrivos shows that their non-dimensional entrained flow rate does begin to fall as inertial terms start to have an effect [2]. Jin and Acrivos recognize that a maximum exists at a capillary number around one.

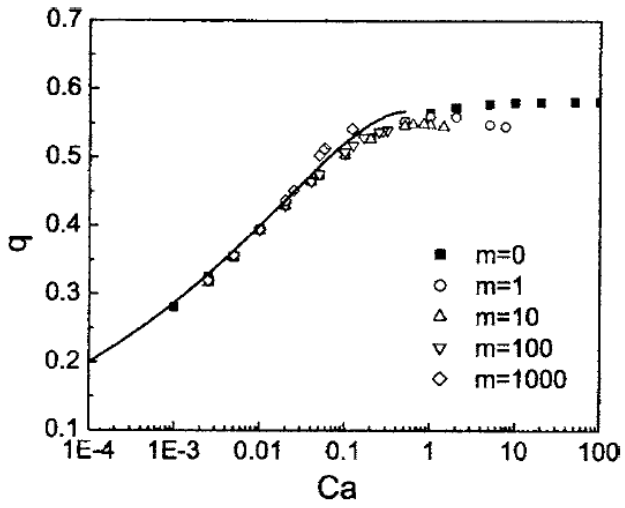

FIG. 5. The flow rate q across the film increases with the capillary number Ca at m=0, 1, 10, 100, and 1000 for the vertical withdrawal case. The points are the results obtained by solving the Navier-Stokes equations and the solid line represents the prediction given by Eq. (2).

Figure 5.2: Flow rate verses capillary number Fig 5. from Jin and Acrivos [2]

5.3 Groenveld Dip Coating Data Comparison

Groenveld constructed a dip coating experiment to test high capillary numbers on Newtonian fluids [6]. The data that was collected was put into excel and the fluid property number was calculated, equation 5.1, where ρ is the density, σ is the surface tension, μ is the viscosity and G is gravity.

$$F_p = \frac{\rho\sigma^3}{\mu^4 G} \tag{5.1}$$

The plug-parabola film equation was run for a series of delta, δ, values ranging from 0.0001 to 0.1 and Weber numbers from zero to 10. The meniscus shape factor, derived in Appendix C and shown below for convenience, was used to determine what value of the Bond number would give a value of one for the meniscus shape factor which corresponds to a liquid pool.

$$a = \frac{\delta}{2Bo}\kappa^2 - \frac{\sin(\alpha) - \delta\frac{dh}{dx}\cos(\alpha)}{\sqrt{1+\delta^2\left(\frac{dh}{dx}\right)^2}} \tag{5.2}$$

Figure 5.3 shows the values of Bond number for each Weber number that will give a meniscus shape factor of one. Straight lines on a logarithmic plot were found at constant values of We, and so Bo is linear in δ with varying slope.

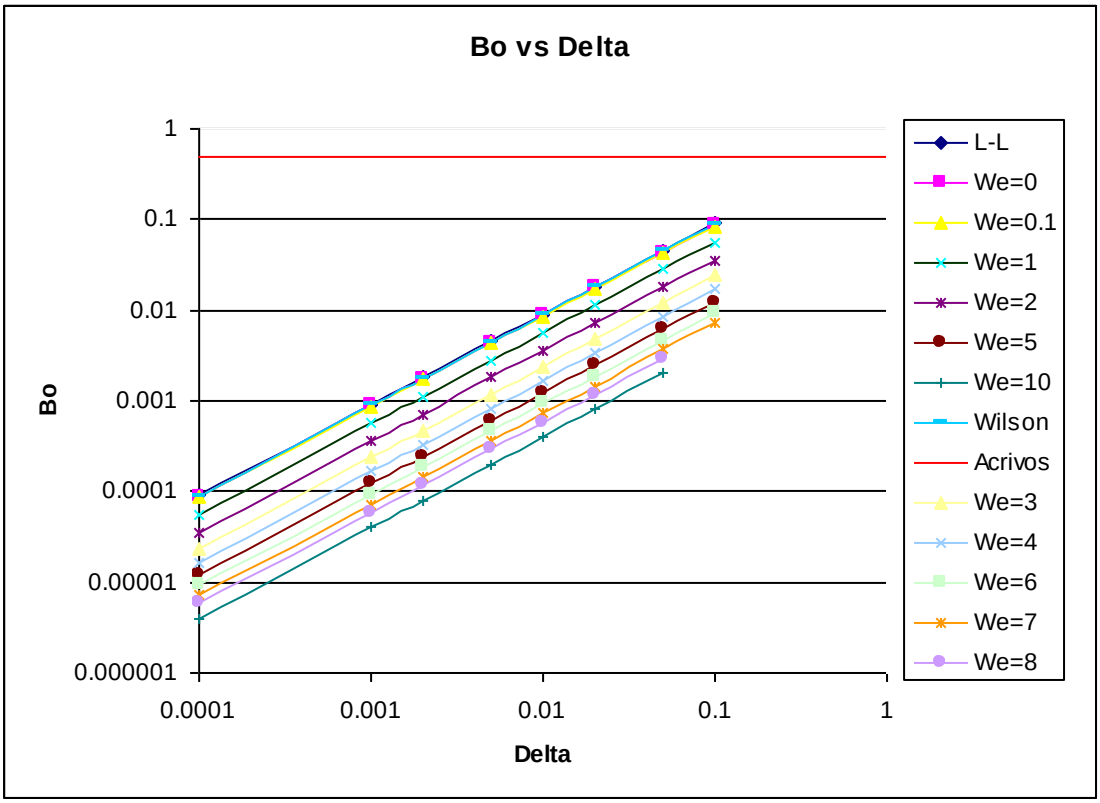

Figure 5.3: Plug-parabola film equation values of Bond number giving meniscus shape factor of one

From the figure, you can see that as the Weber number increases the film thins. Figure 5.3 also shows good agreement between Landau-Levich (L-L), Wilson's corrected equation and low Weber numbers. Wilson corrected the Landau-Levich equation to include gravity and is shown in equation 5.3 [8]. Acrivos from his work found an asymptote for the Bond number. This asymptote states that the bond number cannot be greater than 0.48. The data from the plug-parabola film equation does not cross this asymptote.

$$Bo = \frac{k_1 \delta}{\left[1+\sin(\alpha)\right]}\left\{1-\frac{2k_2}{k_1}\frac{\cos(\alpha)}{\left[1+\sin(\alpha)\right]}\delta\right\} \tag{5.3}$$

$k_1 = 0.94581$

$k_2 = 0.10685$

The slope of each line in figure 5.3 was found using a simple line equation 5.4.

$$Bo=slope*\delta \tag{5.4}$$

The slopes from figure 5.3 matched well with an exponential function. The slopes were used to create a correlation for the data between the Bond number and delta. The constants in equation 5.5 were found using Excel solver and the method of least square reduction.

$$Slope = c\left(e^{-mfitWe} + be^{-aWe^{d}}\right) \tag{5.5}$$

$$a = 0.1013 \quad b = 0.2039 \quad c = 0.75483 \quad d = 1.1578 \quad mfit = 0.560979$$

Equation 5.4 was used with Newton's method to find the Bond number for Groenveld's experimental data. Figure 5.4 shows Groenveld's experimental data and the correlation data. These plots have Wilson's correction, equation 5.3, and Acrivos asymptote plotted as well.

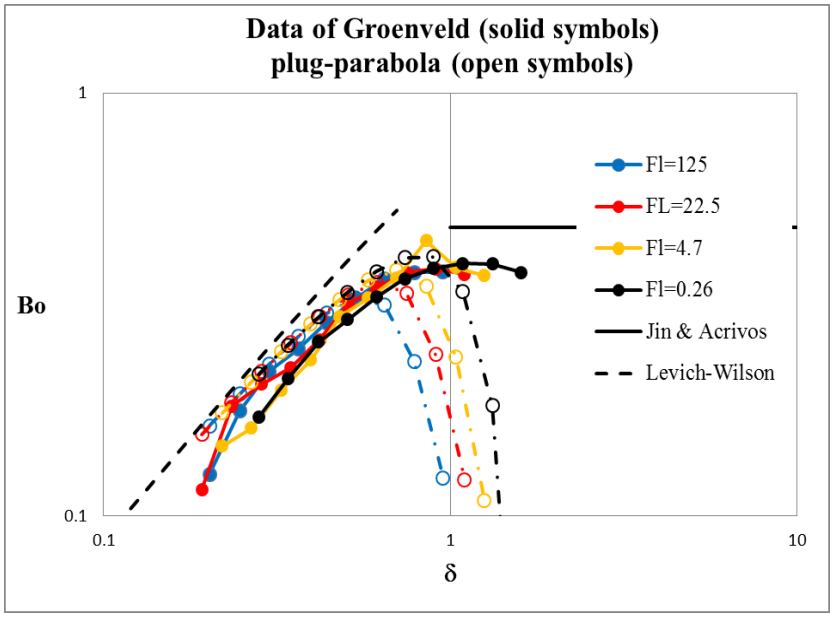

Figure 5.4: Groenveld experimental data and correlation data
Bond number versus delta at various fluid numbers

Figure 5.4 shows that there is a predicted thinning of the film as the speed goes up. The differences between Groenveld's experimental data and the correlation data obtained from the plug-parabola film equation suggest that the plug-parabola shows the effects of inertia early.

5.4 Coyle Slot Coater Meniscus Shape Comparison

Figures 5.5 through 5.7 show the comparison of the plug-parabola film equation, parabola film equation and Coyle's slot coater CFD analysis. Figure 5.5 with no inertia, Re=0, shows good agreement between all three solutions. Recall that the parabola and plug-parabola film equations are identical without inertia. Figures 5.6 and 5.7 at higher Reynolds numbers show that the parabola film equation as well as Coyle's CFD calculation agrees with each other. There is little variation between the data from Coyle and the plug-parabola film equation at Reynolds numbers of 10 and 50 from the solution at Reynolds number of 0. The plug-parabola at higher Reynolds numbers does not agree well with either the slot coater data or the parabola data. The inertia affects become important more quickly within the plug-parabola code.

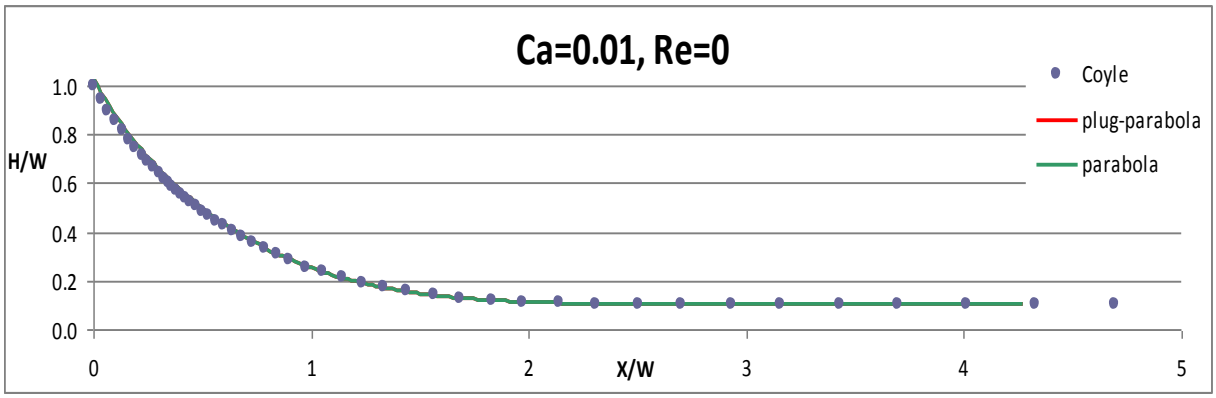

Figure 5.5: Comparison Coyle, plug-parabola and parabola meniscus shapes Re=0, Ca=0.01

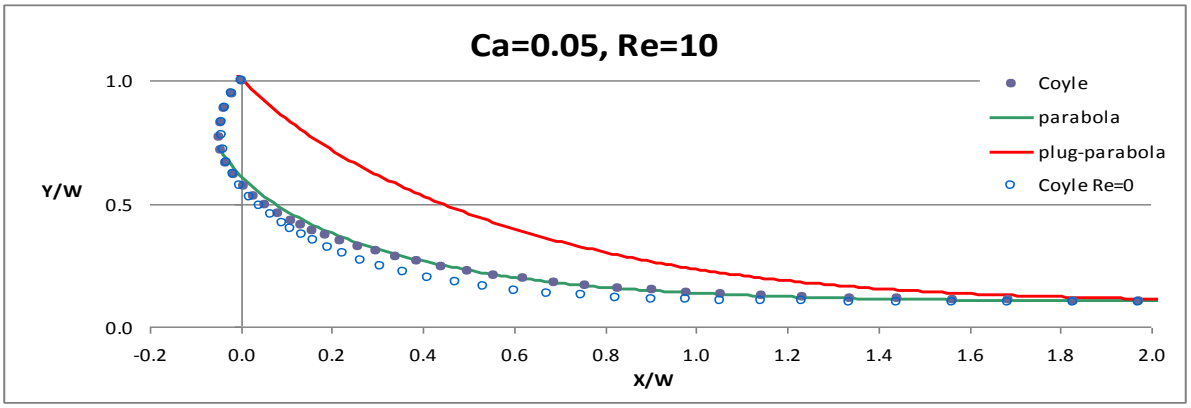

Figure 5.6: Comparison Coyle, plug-parabola and parabola meniscus shapes Re=10, Ca=0.05

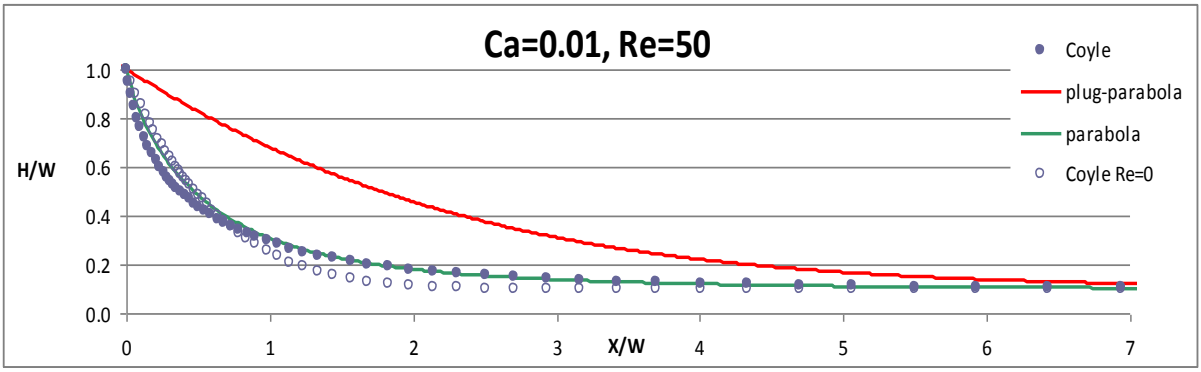

Figure 5.7: Comparison Coyle, plug-parabola and parabola meniscus shapes Re=50, Ca=0.01

5.5 Carvalho Slot Coater Data Comparison

Within Carvalho's paper, it is explained that there is an acceptable coating window that allows for desired coating quality and properties [7]. The low-flow limit is used to describe the maximum and minimum web speed that can be used for a desired film thickness. It is described in Carvalho's paper that the Landau-Levich model can be used to predict this low-flow limit in cases with small capillary numbers.

In Carvalho's work, the low-flow limit was found using his developed numerical model for a variety of property numbers. The equation used to calculate the property numbers is shown in equation 5.6, where ρ is the density, σ is the surface tension, W is the slot gap

and μ is the viscosity. It was found that when the property numbers increase, the effects of the low-flow limit cause a thinner film.

$$P_p = \frac{\rho\sigma W}{\mu^2} \tag{5.6}$$

The low-flow limit for the plug-parabola equation was found for different fluid property numbers, equation 5.7.

$$F_p = \frac{\rho\sigma D}{\mu^2} \tag{5.7}$$

Figure 5.8 shows the maximum gap that can be had with a slot coater. When there is no inertia present and the capillary number is increased, the gap between the slot coater and the substrate must become closer until it reaches a point where it is not physically possible. This affect is shown in figure 5.8 with the yellow points. When designing a slot coating in industry, the W/D is specified and the smallest possible gap is chosen. When W/D is ten, the smallest gap distance happens around a capillary number of 0.03. If the capillary number continues to increase, there will be a point when the maximum coating gap will become possible again as seen in figure 5.8; however, this does not happen in the case with no inertia. The same qualitative relationship between the property number and the maximum gap was found as in Carvalho's data.

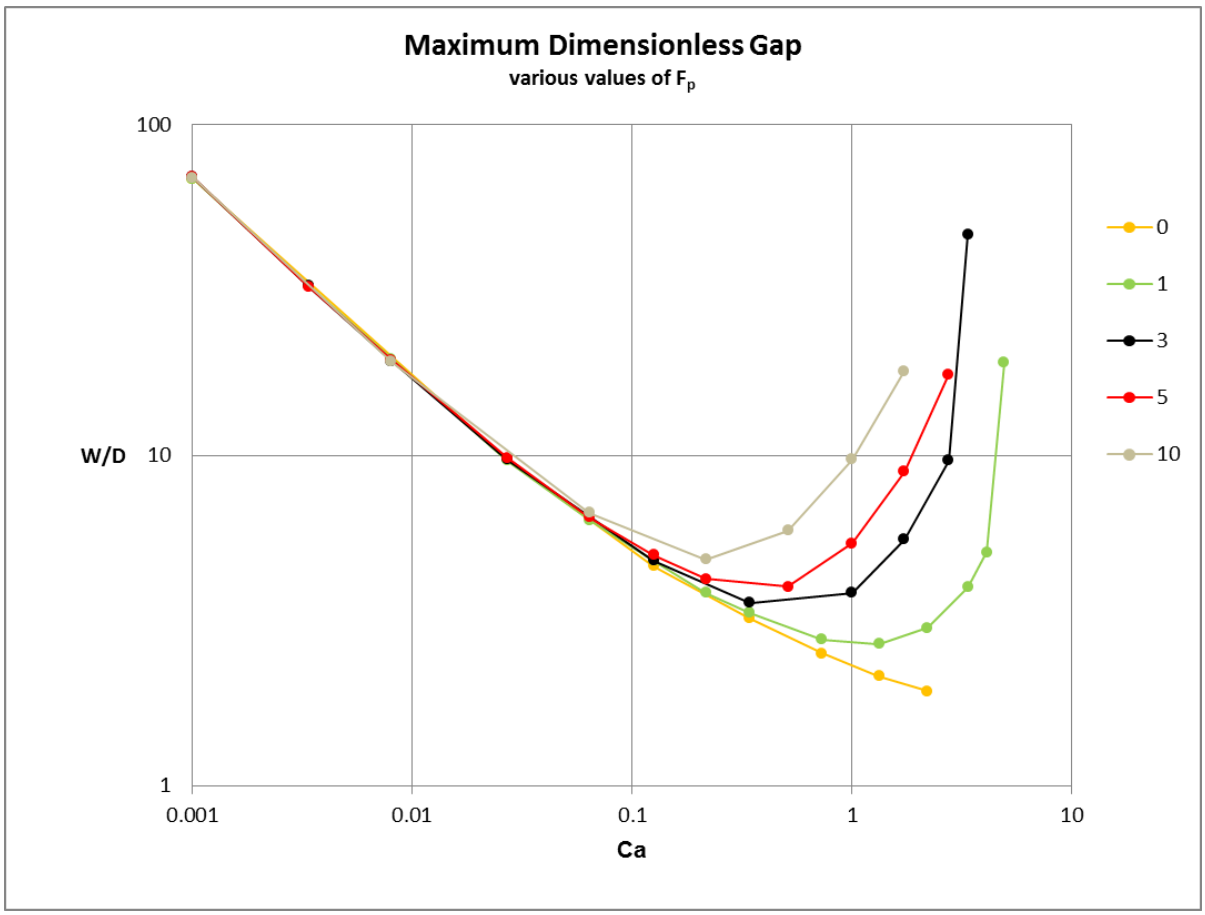

Figure 5.8: Maximum dimensionless gap plug-parabola

Initially when capillary number is increased, the meniscus shape tightens as curvature increases; however, this affect reaches a point where the meniscus shape can no longer reach the lip of the coater. If the capillary number continues to increase, the meniscus will again be able to reach back to the lip of the coater in a tightened shape; now when the capillary number is increased, the meniscus shape relaxes.

Carvalho found this same phenomenon to happen with its numerical model at a fix property number; however, he was unable to obtain results for capillary numbers greater than 1.006. Figure 5.9 shows the effect of increasing capillary number on the meniscus profiles from the plug-parabola film equation and figure 5.10 shows the meniscus profiles within the inertia region. Unlike Carvalho's data,the plug-parabola film equation could find solutions for high capillary numbers.

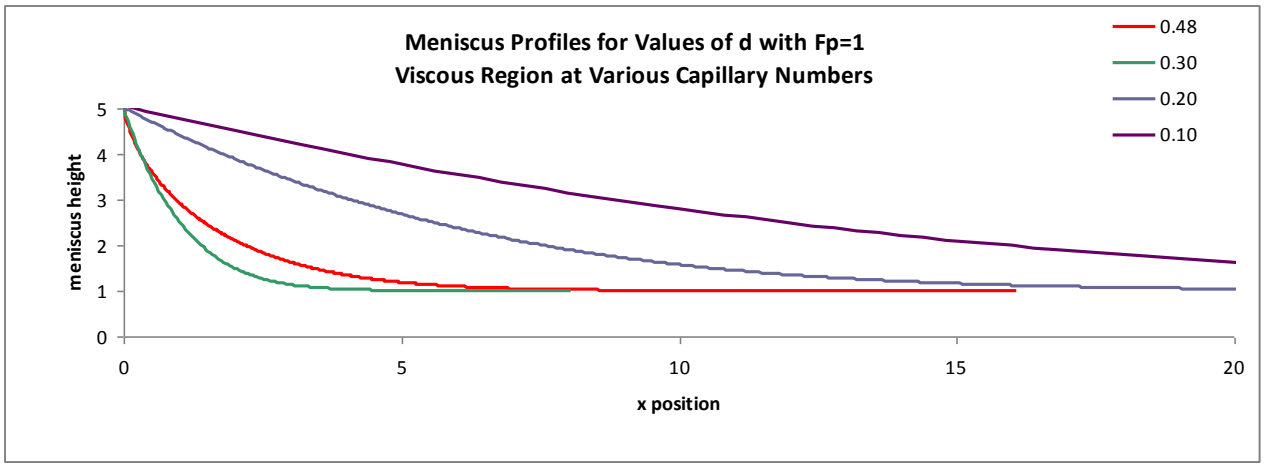

Figure 5.9: Meniscus profiles plug-parabola viscous region

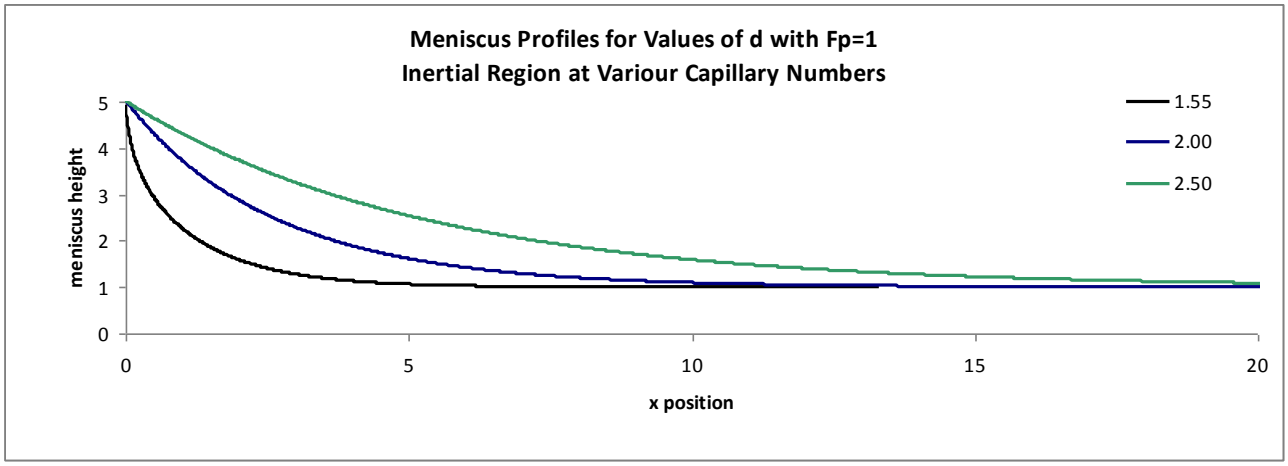

Figure 5.10: Meniscus profiles plug-parabola inertial region

The same trend seen between Coyle's results and the plug-parabola and parabola film equation, section 5.4, can be seen in graphs comparing the meniscus shapes at different capillary numbers and fluid property numbers from Carvalho's data shown in figures 5.11, 5.12, 5.13 and 5.14.

When the Weber number is small, figure 5.11, there is good agreement between Carvalho's data and the parabolic film equation. In figures 5.12, 5.13 and 5.14, there is good agreement between Carvalho's data and the parabola film equation downstream; however, the parabola film equation is unable to reach the same height as Carvalho's data. As the Weber number increases, the discrepancy between the two final heights increases. The final height of the parabola film equation can be solved by looking at the

behavior of the inertia terms in equation 3.4b. As the slope goes to infinity, the dimensionless flow rate goes to one. The inertia term within the brackets must then equal zero giving a value of h=2.5.

Also, seen in figure 5.11 through 5.14, the plug-parabola starts to relax early. This relaxing becomes more apparent as the Weber number increases.

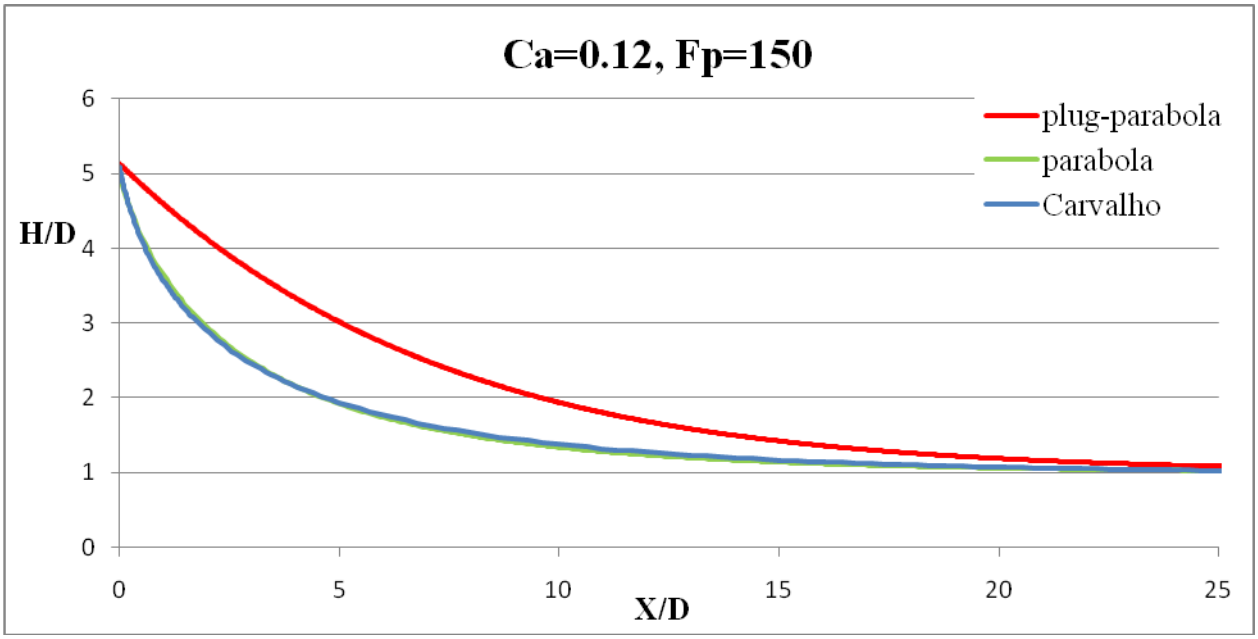

Figure 5.11: Carvalho meniscus shape comparison Ca=0.12, Fp=150, We=9

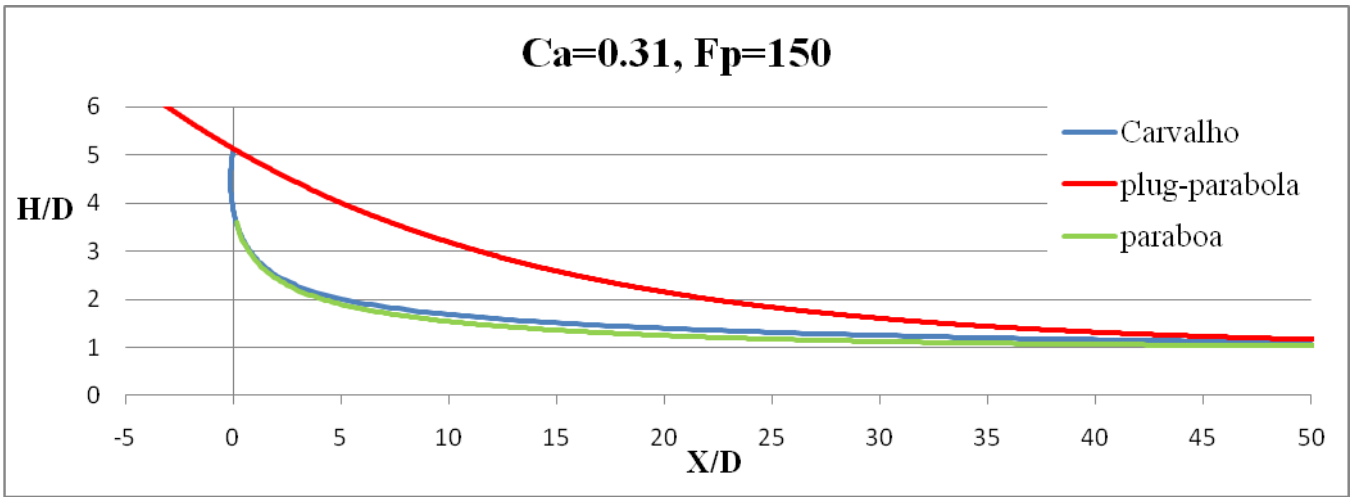

Figure 5.12: Carvalho meniscus shape comparison Ca=0.31, Fp=150, We=31

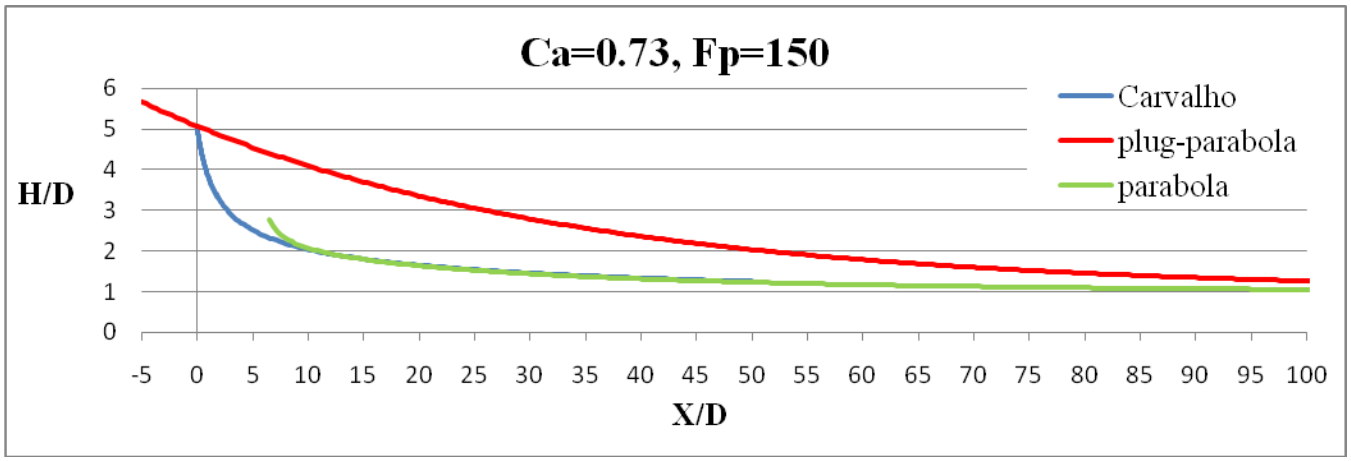

Figure 5.13: Carvalho meniscus shape comparison Ca=0.73, Fp=150, We= 98

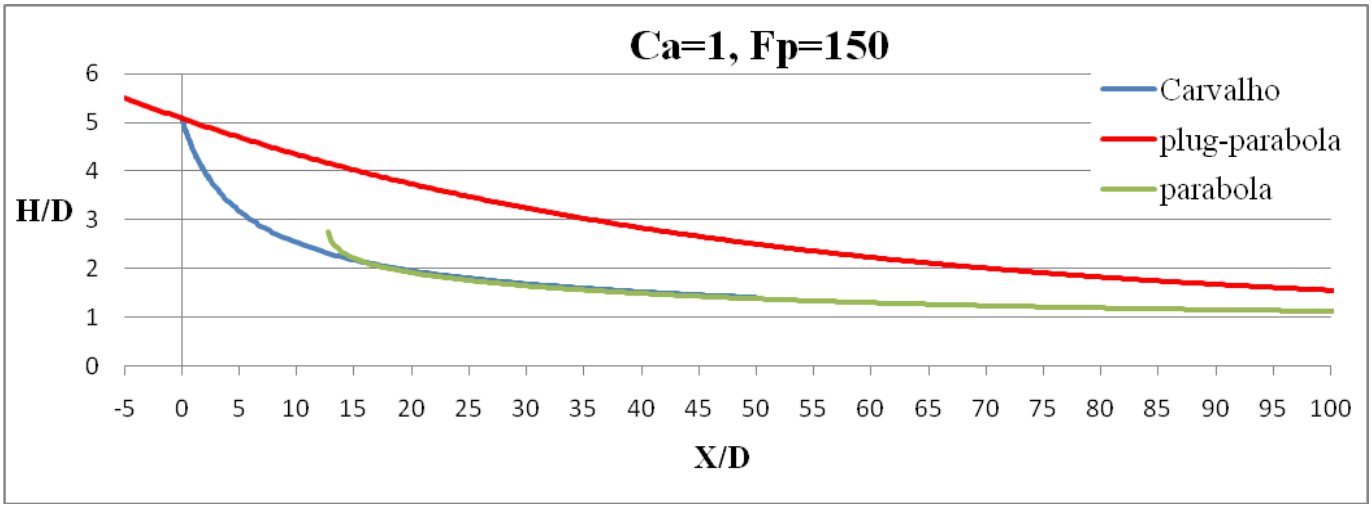

Figure 5.14: Carvalho meniscus shape comparison Ca=1, Fp=150, We=150

Chapter 6
Summary and Recommendations

6.1 Summary

Three test problems were solved to test the assumption that a parabolic velocity profile assumed in creating the film equation causes the deficiency explained in our work. The full set of boundary-layer equations, linearized about the flow far downstream, was solved exactly, without assuming a velocity profile. Fluent was used to solve two fixed boundary models, the wedge and slot die. Results for slot coating were obtained from collaborators of complete free boundary calculations using their own personal CFD codes. These results supported the use of a parabolic velocity profile as a good approximation in the creation of a film equation.

From the numerical analysis in Chapter 4, it was found that a parabolic velocity profile was consistent with the velocity profiles found in the wedge CFD problem, slot CFD problem and boundary later approximation. Although, some authors use higher order polynomials, there was no support found in their work [7].

The comparisons, Chapter 5, show that when there is no gravity there is good agreement between the plug-parabola, parabola and the full boundary-layer approximation. The comparisons done with Groenveld, Carvalho and Coyle's data show that the plug-parabola film equation tends to show the effects of inertia more quickly than the data suggests; however, the parabola equation matches well with the data at moderate Weber numbers. The parabola film equation was shown to limit to a maximum value of h of 2.45 as Weber number and the slope became large. Thus the parabola film equation does not show meniscus relaxation.

The plug-parabola equation was found to be able to show the relaxation and tightening of the meniscus in a slot coater. The plug-parabola equation tends to relax too soon; however, relaxation is not seen in the parabola film equation at all. It was also found that

the plug-parabola can produce film profiles at high capillary numbers which the parabola code could not. The plug-parabola from Groenveld's data showed that there was a thinning of the film as the speed goes up.

The plug-parabola is able to limit back to a static meniscus unlike the parabola film equation. However, the plug-parabola had two unexpected but desirable features. It showed meniscus relaxation as inertia became dominant and the ability to compute film profiles at high Weber and capillary numbers. So, it predicts the extension of the coating window that Carvalho found. That effect is well known to coating practitioners and is important because it enables the coating of thinner films at high speed [7, Ken Ruschak private communication].

6.2 Recommendations

There were difficulties associated with using Fluent for the numerical simulations. The major difficulty was encountered when a free surface problem had to be solved. The problem was modeled after the slot coater problem in section 4.3 with the major difference being that the meniscus shape was not determined by the parabola code but left to deform as the solution was solved. Fluent was unable to solve this problem. The Fluent online support was also unable to give a reasonable way to solve this free surface problem. Fluent uses the volume of fluid method (VOF) to solve free surface problems and was unable to handle this configuration within Fluent. Additionally, this method was later shown by Kam Ng using Flow 3D to be unable to solve this problem accurately.

The plug-parabola film equation shows desirable characteristics including its ability to limit to a static meniscus, its ability to show the relaxation of the meniscus as inertia becomes dominant, and its ability to produce solutions at high capillary numbers; however, the plug-parabola equation also showed that the inertial affects become predominant early compared with the literature data and the parabola film equation.

It is recommended that a combination of the plug-parabola and parabola film equation be created with weighting factors used to control the plug-parabola equation and the inertial affects. The weighting would favor the parabola film equation when the film is thin and the plug-parabola film equation when the film is thick. So, the proven accuracy of the parabola film equation for thin films would be preserved and also the ability to limit to a static meniscus at large film thicknesses. Also, more data needs to be generated from finite element solutions at high inertia for comparisons with the film equations. Of particular interest is the high inertia condition where the parabolic film equation was found to lose accuracy. Another recommendation would be to explore the predications of the plug-parabola film equation for other coating flows for its ability to capture the known behavior of high speed coating.

References

[1] H. S. Kheshgi, "Profile Equations for Film Flows at Moderate Reynolgs Numbers," *AIChE Journal,* vol. 35, pp. 1719-1727, 1989.

[2] B. Jin, A. Acrivos, and A. Munch, "The drag-out problem in film coating," *Physics of Fluids,* vol. 17, pp. 103603-12, 2005.

[3] S. F. Kistler and P. M. Schweizer, *Liquid film coating : scientific principles and their technological implications*. London; New York: Chapman & Hall, 1997.

[4] S. J. Weinstein and K. J. Ruschak, "Dip coating on a planar non-vertical substrate in the limit of negligible surface tension," *Chemical Engineering Science,* vol. 56, pp. 4957-4969, 2001.

[5] K. J. Ruschak, "Flow of Falling Film into a Pool," *AIChE Journal,* vol. 24, pp. 705-709, 1978.

[6] P. Groenveld, "High capillary number withdrawal from viscous Newtonian liquids by flat plates," *Chemical Engineering Science,* vol. 25, pp. 33-40, 1969.

[7] M. Carvalho and H. Kheshgi, "Low-Flow Limit in Slot-Coating: Theory and Experiments," *AIChE,* vol. 46, pp. 1907-1917, 2000.

[8] S. D. R. Wilson, "The drag-out problem in film coating theory," *Journal of Engineering Mathematics,* vol. 16, pp. 209-221, 1982.

[9] K. S. Kheshgi, S. F. Kistler, and L. E. Scriven, "Rising and Falling Film Flows: Viewed From a First-Order Approximation," *Chemical Engineering Science,* vol. 47, pp. 683-694, 1992.

[10] D. Marques, V. Costanza, and R. L. Cerro, "Dip Coating at Large Capillary Numbers: an Initial Value Problem," *Chemical Engineering Science,* vol. 33, pp. 87-93, 1978.

[11] C. V. Subbaraman and W. L. Wilkinson, "Free coating of A Newtonian liquid onto a vertical surface," *Chemical Engineering Science,* vol. 29, pp. 389-396, 1974.

[12] J. P. Kizito, Y. Kamotani, and S. Ostrach, "Experimental free coating flows at high capillary and Reynolds number," *Experiments in Fluids,* vol. 27, pp. 235-243, 1999.

[13] H. I. Andersson, "The Momentum Integral Approach to Laminar Think-Film Flow," *ASME,* vol. 48, pp. 7-13, 1987.

[14] N. Annapurna and G. Ramanaiah, "A Unified Treatment of Drainage, Withdrawal, and Post withdrawal Drainage with Inertial Effects," *AIChE Journal,* vol. 5, pp. 940-942, 1976.

[15] R. L. Cerro and L. E. Scriven, "Rapid Free Surface Film Flows. An Integral Approach," *Industrial and Engineering Chemistry Fundamentals,* vol. 19, pp. 40-50, 1980.

[16] M. N. Esmail and R. L. Hummel, "Nonlinear Theory of Free Coating Onto a Vertical Surface," *AIChE,* vol. 21, pp. 958-965, 1975.

[17] B. G. Higgins and L. E. Scriven, "Interfacial Shape and Evolution Equations for Liquid Films and Other Viscocapillary Flows," *American Chemical Society,* vol. 18, pp. 208-215, 1979.

[18] R. E. Hildebrand and J. A. Tallmadge, "Direct Evaluation of Withdrawal Equations," *AIChE Journal,* vol. 14, pp. 660-661, 1968.

[19] L. Landau and B. Levich, "Dragging of a Liquid by a Moving Plate," *ACTA Physicochimica URSS,* vol. 17, pp. 42-54, 1942.

[20] K. J. Ruschak, "Coating Flows," *Ann. Rev. Fluid Mech.,* vol. 17, pp. 65-89, 1985.

[21] A. J. Soroka and J. A. Tallmadge, "A Test of the Inertial Theory for Plate Withdrawal," *AIChE,* vol. 17, pp. 505-508, 1971.

[22] S. J. Weinstein and K. J. Ruschak, "Coating Flows," *Annual Review of Fluid Mechanics,* vol. 36, pp. 29-53, 2004.

[23] S. J. Weinstein and K. J. Ruschak, "On the mathematical structure of think film equations containing a critical point," *Chemical Engineering Science,* vol. 54, pp. 977-985, 1999.

[24] B. G. Higgins, W. J. Silliman, R. A. Brown, and L. E. Scriven, "Theory of Meniscus Shape In Film flows. A Synthesis," *Industrial and Engineering Chemistry Fundamentals,* vol. 16, pp. 393-401, 1977.

[25] P. Tanguy, M. Fortin, and L. Choplin, "Finite Element Simulation of Dip Coating, I: Newtonian Fluids," *International Journal For Numerical methods In Fluids,* vol. 4, pp. 441-457, 1984.

[26] H. I. Andersson, "On Integral Method Predictions of Laminar Film Flow," *Chemical Engineering Science,* vol. 39, pp. 1005-1010, 1983.

Appendix A

Fluent Wedge Problem

This appendix contains a detailed description of the problem layout and steps to create the Gambit mesh as well as the Fluent solution.

A.1 Derivation of the Velocity Profile

The problem's geometry and boundary conditions are shown in Chapter 4 section 2. The boundary conditions for the inlet fully developed velocity profile are shown below in equation A.2. The boundary condition at the moving wall was no slip boundary condition. A no shear boundary condition was chosen for the top wall of the wedge. At the inlet, the integral of the velocity is equal to the flow rate. A second order velocity profile was chosen and was solved for using the boundary conditions shown in A.2 resulting in equation A.4. The constants A, B and C are solved for and are entered into the velocity profile with the dimensionless group shown in A.4.

$$u = S \quad \text{at} \quad y = 0 \tag{A.2a}$$

$$\frac{du}{dy} = 0 \quad \text{at } y = h \tag{A.2b}$$

$$\int_0^H u\,dy = q \tag{A.2c}$$

$$u = S\left[1 - \frac{3(\lambda - 1)}{2H^2}y^2 - 2Hy\right] \tag{A.3}$$

$$\lambda = \frac{q}{SH} \tag{A.4}$$

A.2 Gambit Mesh Generation

The four (x, y) coordinates found in figure 4.1 are entered into Gambit using the insert point tool. The steps for creating the faces to be meshed are standard; however, the mesh for the wedge problem was created using a sizing function. The sizing function was created to attach to the face, start at all edges with a start size of 0.05 for 20 nodes across the inlet and a growth rate of 1.1. A pave mesh was selected within the meshing options along with the sizing function.

A.3 Velocity Profile User Defined Function Code

The general C code used to import the inlet velocity profile into Fluent is listed below. Within the user defined function (udf) code, the values for the inlet height, bottom wall speed and the dimensionless flow rate are entered. The file then is saved as a .c file and can be imported into Fluent.

```
/** Fuent Directories **/
#include "udf.h"
#include "mem.h"
/** Parameter Constants **/

#define wallspeed 200.0

#define flowrate 0.01
#define height .05

/*****************  Inlet Velocity Profile in Wedge  *****************/
DEFINE_PROFILE(inlet_velocity_profile,thread,position)
{
  real x[ND_ND];
  real y;
  face_t f;

  begin_f_loop(f,thread)
    {
      F_CENTROID(x,f,thread);
      y = fabs(x[1]);
```

```
    F_PROFILE(f,thread,position) = wallspeed*(1 - (((3*(flowrate-
1))/(2*(height*height)))*((y*y)-(2*height*y))));
   }
  end_f_loop(f,thread)
}
```

A.4 Fluent Solver Set-up

The Fluent 2-D double precisian solver setup is standard with a few exceptions explained within this section of the Appendix. The 2-D mesh is imported into Fluent. The udf c code must be interpreted into Fluent. To interpret the udf, use the following steps: select define, user defined, functions, interpreted and then select your file and click interpret button.

The boundary condition selection and specifications are standard for all the boundary conditions except for the inlet. The inlet condition must be specified as the x component of velocity being udf and the y component of velocity being zero. The locations of the lines used to extract data are shown in table A.1. When exporting the data from Fluent after the run has been completed, the data must be exported as an ACSI file with a .cvs extension.

Location From Outlet	x (m)	y (m)
0.5	0.075	0.03
0.75	0.0375	0.06
0.875	0.01875	0.055
Inlet	.15	.05

Table A.1: Locations of Data collection lines

A.5 Plotting Fluent Data MatLab Code

The following code was used to create a plot of the Fluent data once it has been imported into MatLab using the code in Appendix G. The data is scaled to go from zero to one on both axes in order to make the velocity profiles at each line comparable.

```
function
[Values]=linemake(xcoordinate,ycoordinate,velocitymagnitude,xvelocity,y
velocity)

%Finding number of values for the problem
%disp('*********Did you Change Wall Speed in linemake
function?*********')

lengthx=length(xcoordinate);
count=2;
S=200;    %wall speed (m/s)

y=zeros(100,1);
velcheck=zeros(100,1);

Values=struct('linex',{},'liney',{},'velmag',{},'xvel',{},'yvel',{});

 s=1; %counter

%%%%Creating Velocity Check to be graphed%%%%

%filling first row of matrix
Values(1).linex(1,1)=xcoordinate(1,1);
Values(1).liney(1,1)=ycoordinate(1,1);
Values(1).velmag(1,1)=velocitymagnitude(1,1);
Values(1).xvel(1,1)=(-1)*xvelocity(1,1)/S;
Values(1).yvel(1,1)=yvelocity(1,1);


%Creating an array that holds the different line values for x,y and
%velocity
 for i=2:lengthx
    res=abs(xcoordinate(i,1)-xcoordinate(i-1,1));

       if res > 1e-8
        s=s+1;
        count=1;
       end

   Values(s).linex(count,1)=xcoordinate(i,1);
   Values(s).liney(count,1)=ycoordinate(i,1);
   Values(s).velmag(count,1)=velocitymagnitude(i,1);
   Values(s).xvel(count,1)=(-1)*xvelocity(i,1)/S;
   Values(s).yvel(count,1)=yvelocity(i,1);

   count=count+1;
 end
 %Mapping values so xvel goes from 0 to 1 as well as y
 lengthval=length(Values);

 for i=1:lengthval
     maxy=max(Values(1,i).liney);
```

```
     lengthy=length(Values(1,i).liney);
     for k=1:lengthy
         Values(1,i).liney(k,1)=Values(1,i).liney(k,1)/maxy;
     end
 end

%%%%Creating Plots%%%%%

figure(1)
plot(Values(1,4).liney(:,1),Values(1,4).velmag(:,1),'b :',...
    Values(1,3).liney(:,1),Values(1,3).velmag(:,1),'m :',...
    Values(1,2).liney(:,1),Values(1,2).velmag(:,1),'r :',...
    Values(1,1).liney(:,1),Values(1,1).velmag(:,1),'k :')

     title('Velocity Magnitude vs y Postion n/ Bottom Wall Moving to
right')
     xlabel('y position (m)')
     ylabel('Velocity Magitude (m/s)')
     legend('Location (0.13125,0.045)','Location (0.1125,0.04)',...
            'Location (0.075,0.03)','location','NorthEast')
 End
```

Appendix B

Fluent Slot Die Problem

This appendix contains a detailed description of the problem layout and steps to create the Gambit mesh as well as the Fluent solution.

B.1 Derivation of the Velocity Profile

The boundary conditions for the inlet fully developed velocity profile are shown below in equation B.1. Boundary at the moving wall was chosen to be a no slip boundary condition. A no slip boundary condition was chosen for the top wall of the slot entrance. At the inlet, the integral of the velocity is equal to the flow rate. A second order velocity profile was chosen and was solved for using the boundary conditions shown in B.1. The constants A, B and C are solved for and entered into the velocity profile. The final velocity profile found is shown in equation B.2.

$u = S$ at $y = 0$ (B.1 a)

$u = 0$ at $y = W$ (B.1 b)

$$\int_0^W u dy = q = SD \quad \text{(B.1 c)}$$

$$u = S\left[\frac{6}{W^2}\left(\frac{1}{2} - \frac{D}{W}\right)\left(y^2 - Wy\right) - \frac{y}{W} + 1\right] \quad \text{(B.2)}$$

B.2 Gambit Mesh Generation

The process for creating the Gambit mesh of the slot die problem is standard; however, the mesh of the slot die problem was created using a sizing function and the points for the meniscus shape/curve are imported from the MatLab code shown in B-4. Vertex points

given from the MatLab code should be imported into Gambit by going to File, Import, Vertex Data and selecting the file name where the points were saved. In order to create a smooth edge on the curved portion of the vertex data, right click on the EDGE button from the drop down menu, select NURBS option, select the curved portion of the vertex data, and choose the approximate with a tolerance of zero. The sizing function created for the mesh was source all edges, attachment face, start size W/20, Growth Rate was 1.1 and the max size was 1. A pave mesh was selected within the meshing options along with the sizing function.

B.3 Gambit Mesh Generation MatLab Code

This code solves the parabola film equation derived in Appendix D. The inputs that are selected by the user are the Reynolds number, capillary number and the dimensionless flow rate. The output of this code is the points needed to create the mesh within gambit for the slot die problem. Once the code has run, open the variable named "points" from the workspace, copy this data and paste it into a notepad document, save the document with a .dat extension.

The point generation code uses two subprograms or functions root and fn. The fn function holds the set of first order ODEs that is solved from the parabola film equation. The root function solves for the roots of the liberalized form of the parabola film equation derived in Appendix D.

Point Generation Code

This code solves the third order film equation by converting it to 3 first order equations and using ode45

```
%ODE Film Equation Solver Solt Die Point Generator%
%Outputs x,y,z points for slot die Gambit mesh
%Current as of 4-15-10
%-------------------------------------------------------------------------%
clc

%Constants

global Bo Alpha Re C D W
```

```
%Choosen Values for problem
Bo= 0;                              %Bond Number
Re= 50;                            %Reynolds Numbe
Ca=0.01;                            %capillary number
FinalFilm=0.01;                        %final Film thickness
C= 1;                             %Flow Rate dimensionless
D= Ca^(1/3);                        %Delta =(capillary number)^1/3

Alphadeg= 0;                        %Angle of incline degrees
Alpha=Alphadeg*(pi()/180);          %Convert Alpha to radians

W=0;                                %Constant to solve for L-L or full
                                    %Equation can be 0(L-L) or 1

K=.0001;                            %Constant from solving the diff
equation
n=0;                                %Counting Constant
t=100;                              %Value of dh/dx to stop code

  disp(Bo);
  disp(D);

%-------------------------------------------------------------------------%

if Bo*cos(Alpha)>1
    disp('Bocos(Alpha) is greater than 1')
end

%%Root Finder program Newton's Method %%
%F(m)=m^3+m*[Bo*d*sin(Alpha)-(Re*d/5)*(1-6C^2)]+3*(1-Bocos(Alpha))

m=-1*(3*(1-Bo*cos(Alpha)))^(1/3);  %Original guess for root

%Newton's Method Finding Roots

kk=root(m); %Calling function root
F=kk^3+kk*(Bo*D*sin(Alpha)-((Re*D/5)*...
        (1-6*C^2)))+3*(1-Bo*cos(Alpha));
    disp(F);

%-------------------------------------------------------------------------%

%%Film Equation Solution%%

%Creating a System of first order ODEs
    %y1=h y2=h' y3=h''
    %System of equations
    %y(1)'=y(2), y(2)'=y(3),
    %y(3)'=[1+d^2*y(2)^2]^(3/2)*{[Re*d/5*y(1)]*[1-(6*C^2/y(1)^2)]*...
    %y(2)+(3*d^2*y(2)*y(3)^2)/[1+d^2*y(2)^2]^(5/2)-Bo*d*sin(Alpha)*...
    %y(2)-Bo*cos(Alpha)+3*(y(1)-C)/y(1)^3}

while n>=0
y0=[1+K;-m*K;K*m^2];         %Initial conditions
```

```
xspan=[0 10+n];                %X interval

[x,y]=ode45(@fn,xspan,y0);   %Diff equation solution

s=length(y);                 %Finding the size of y for loop

%Creating matrix constants
tdriv=zeros(s,1);
crv=zeros(s,1);
dcrv=zeros(s,1);
dstaticm=ones(s,1);
a=zeros(s,1);
aa=zeros(s,1);
h=zeros(s,1);

for i=1:s
%Calculation of d3h/dx3
G =(Re*D/5*y(i,1))*((1-(6*C^2/y(i,1)^2))*y(i,2));

R =(W*3*D^2*y(i,2)*y(i,3)^2)/((1+D^2*y(i,2)^2)^(5/2));

A = -Bo*D*sin(Alpha)*y(i,2)-Bo*cos(Alpha)+((3*(y(i,1)-C))/y(i,1)^3);

tdriv(i,1)=((1+W*D^2*y(i,2)^2)^(3/2))*(G+R+A);

%Curvature equation
crv(i,1)=y(i,3)/((1+W*D^2*y(i,2)^2)^(3/2));

%Derivative curvature equation
dcrv(i,1)=(-1*tdriv(i,1)/((1+W*D^2*y(i,2)^2)^(3/2)))+ R;

    if Bo ~= 0
        %Bo does not equal zero

        %Derivative Static meniscus equation
        P=D*y(i,3)*cos(Alpha)*((1+D^2*y(i,2)^2)^.5);

        Q=((sin(Alpha)-D*y(i,2)*cos(Alpha)))*D^2*y(i,3)*-1*...
            y(i,2)*((1+D^2*y(i,2)^2)^(-1/2));

        dstaticm(i,1)=((D/Bo)*crv(i,1)*dcrv(i,1))-((P-
Q)/(1+D^2*y(i,2)^2));

        %Calculation Alpha ()
        aa(i,1)=(sin(Alpha)-
D*y(i,2)*cos(Alpha))/((1+D^2*y(i,2)^2)^(1/2));
        a(i,1)=((D/Bo*2)*crv(i,1)^2)- aa(i,1);

    end

%solving of h=1+h'
h(i,1)=1+K*exp(-m*x(i,1));
```

```
end

n=n+1; %change time of integration

%disp(num2str(n));

 %Convergence Check---1st derivative becomes too large

       %Convergence Check---1st derivative becomes too large
     for i=i:s
        v = y(i,2);
         if  v > t
             if Bo~=0
                 if abs(dstaticm(i,1)) < 0.001 %Checking the 1st
derivative value
                   n=-1;                    %Value of n to break while
loop
                   break                    %Breaks for Loop
                 end
             else
                 %Case when Bo=0
                 if dcrv(i,1) <.001
                     n=-1;
                     break
                 end
             end
          end
     end

             if n>300
                 n=-1;
             end
end

%%------Creating Plots and Points for Meshing of Meniscus shape--------
-%%
s=length(h);
counth=0;
countstart=0;

for i=1:s
    if y(i,1)<(1+FinalFilm)
        countstart=countstart+1;
    end

    if y(i,1)<12 %Checking to see if Value of h is great than 10X 1.001
    counth=counth+1;
    end

end

rowstart=countstart+1;
```

```
FFD=FinalFilm/D;
Lpoints=counth-rowstart; %finding total number of rows in points

points=zeros(Lpoints+3,3);
rowcount=0;
%Constants to translate x coordinates to have x=0 at inlet

for k=rowstart:counth
rowcount=rowcount+1;
points(rowcount,1)=1000*(x(k,1)*FFD);  %x coordinates
points(rowcount,2)=1000*(y(k,1)*FinalFilm);  % y coordinates
points(rowcount,3)=0.0;     %z coordinates
end

%Adding points for entrance length

     %inlet top right
    points(Lpoints+1,1)=points(Lpoints,1);
    points(Lpoints+1,2)=100;
    points(Lpoints+1,3)=0;

    %inlet top left
    points(Lpoints+2,1)=points(Lpoints,1)+100;
    points(Lpoints+2,2)=100;
    points(Lpoints+2,3)=0;

    %inlet bottom
    points(Lpoints+3,1)=points(Lpoints+2,1);
    points(Lpoints+3,2)=0;
    points(Lpoints+3,3)=0;

    %outlet
    points(Lpoints+4,1)=points(1,1);
    points(Lpoints+4,2)=0;
    points(Lpoints+4,3)=0;

%zeroing points so inlet is at x=0 and shape goes in positive x
direction

pointlength=length(points);
distancex=points(pointlength-1,1)-points(1,1);
setzero=1000*(x(rowstart,1)*FFD);

for u=1:pointlength
    points(u,1)=(-1)*(points(u,1)-distancex-setzero);
end

%outlet making outlet 5*D to ensur plug flow
    points(Lpoints+4,1)=points(pointlength,1)+(5*points(1,2));
    points(Lpoints+4,2)=0;
    points(Lpoints+4,3)=0;

    points(Lpoints+5,1)=points(Lpoints+4,1);
    points(Lpoints+5,2)=points(1,2);
    points(Lpoints+5,3)=0;
```

```
figure(5)
plot(points(:,1),points(:,2)); %x x-axis
title('Film Equation Solution h')
xlabel('x')
ylabel('y( )')
```

Root Function

This solves the third order polynomial using Newton's method to get the relaxation parameter m.

```
function [ m1 ] = root( m )
%%Root Finder program Newton's Method %%
%1-8-10
%ODE solution/Classic Film Equation%
%F(m)=m^3+m*[Bo*d*sin(Alpha)-(Re*d/5)*(1-6C^2)]+3*(1-Bocos(Alpha))

global Bo Alpha Re C D
error= 0.001;        %Setting error constant

    while error>1e-12
    %m1 is new guess from solving m
        F=m^3+m*(Bo*D*sin(Alpha)-((Re*D/5)*...
            (1-6*C^2)))+3*(1-Bo*cos(Alpha)); %Guess plugged in F(m)

         Fp=3*m^2+(Bo*D*sin(Alpha)-((Re*D/5)*(1-6*C^2))); %F'(m) with
guess

        m1=m-(F/Fp);                  % Finding new root
        error=abs(1-(m/m1));          % Finding Error
        m=m1;                         % Replacing Old Guess
    end

disp('Root By Newton''s Method')
disp( num2str(m1))
end
```

fn function: System of first order ODEs

```
function dhdx = fn(x,y)
%1-8-10
global Bo Alpha Re C D W

%Equations in from of 3 separate equations for ODE solver

dhdx=[y(2);y(3);((1+W*D^2*y(2)^2)^(3/2))*((Re*D/5*y(1))*...
    (1-(6*C^2/y(1)^2))*y(2)+(W*3*D^2*y(2)*y(3)^2)...
    /((1+D^2*y(2)^2)^(5/2))- Bo*D*sin(Alpha)*y(2)- Bo*cos(Alpha)...
    +(3*(y(1)-C))/y(1)^3)];

 end
```

B.4 Velocity Profile User Defined Function Code

The general C code used to import the inlet velocity profile into Fluent is listed below. Within the user defined function (udf) code, the values for the wall speed, final film thickness (finalheight) and the entrance height W (height). The file then is saved as a .c file and can be imported into Fluent.

```
/** Fuent Directories **/
#include "udf.h"
#include "mem.h"

/** Parameter Units m and m/s **/

/** Parameter Constants **/

#define wallspeed 0.00025

#define finalheight 0.1

#define height 0.01

/*****************   Inlet Velocity Profile in Slot   *****************/
DEFINE_PROFILE(inlet_velocity_profile,thread,position)
{
  real x[ND_ND];
  real y;
  face_t f;

          real A;
          real B;
          real C;

          A=((6.*wallspeed)/(finalheight*finalheight))*((1./2.)-(height/finalheight));
          B= -1.*(((6.*wallspeed)/(finalheight))*((1./2.)-
(height/finalheight))+(wallspeed/finalheight));
          C=wallspeed;

  begin_f_loop(f,thread)
    {
      F_CENTROID(x,f,thread);
      y = x[1];
```

```
        F_PROFILE(f,thread,position)= A*y*y+B*y+C;
    }
  end_f_loop(f,thread)
}
```

B.5 Fluent Solver Set-up

The Fluent 2-D double precision solver set-up is standard with a few exceptions explained within this section of the Appendix. The 2-D mesh is imported into Fluent. The mesh must be scaled within Fluent by 0.001. The data points imported into Gambit were scaled to increase the speed of meshing within Gambit. The udf c code must be interpreted into Fluent. To interpret the udf, use the following steps: select define, user defined, functions, interpreted and then select your file and click interpret button. The boundary condition selection and specifications are standard for all the boundary conditions except for the inlet. The inlet condition must be specified as the x component of velocity being udf and the y component of velocity being zero. The locations of the lines used to extract data are placed at the inlet, seven eighths, three quarters, five eighths, one half, nine twentieths, three tenths, and one quarter from the inlet. When exporting the data from Fluent after the run has been completed, the data must be exported as an ACSI file with a .cvs extension.

B.6 Plotting Fluent Data MatLab Code

The following code was used to create plot of the Fluent data once it has been imported into MatLab using the code in Appendix G. The data is scaled to go from zero to one on both accesses in order to make the velocity profiles at each line comparable.

```
function
[Values,wallspeed,Values2]=linemake_map(xcoordinate,ycoordinate,velocit
ymagnitude,xvelocity,yvelocity)
disp('Did you change Wallspeed with new Re number?')
%Finding number of values for the problem
lengthx=length(xcoordinate);
count=2;
```

```
%q=1;    %flow rate
w=0.1; %inlet height (m) constant for all Re
D=.01; %final Film thickness (m) constant

y=zeros(100,1);
velcheck=zeros(100,1);

bottomwallspeed=2.5; %(cm/s)

wallspeed=bottomwallspeed*0.01; %(m/s)
S=wallspeed;
Values=struct('linex',{},'liney',{},'velmag',{},'xvel',{},'yvel',{});
Values2=struct('liney',{},'xvel',{});

s=1; %counter
%%%%Creating Velocity Check to be graphed%%%%
    for i=1:20
    y(i,1)=(i-1)*.0005;
    velcheck(i,1)=S*((((6/(w^2))*((1/2)-(D/w)))*(y(i,1)^2-(w*y(i,1))))-
(y(i,1)/w)+1);
    end



%filling first row of matrix
Values(1).linex(1,1)=xcoordinate(1,1);
Values(1).liney(1,1)=ycoordinate(1,1);
Values(1).velmag(1,1)=velocitymagnitude(1,1);
Values(1).xvel(1,1)=xvelocity(1,1);
Values(1).yvel(1,1)=yvelocity(1,1);

%Creating an array that holds the different line values for x,y and
%velocity

 for i=2:lengthx
    res=abs(xcoordinate(i,1)-xcoordinate(i-1,1));
```

```
       if res > 1e-3
        s=s+1;
        count=1;
       end

   Values(s).linex(count,1)=xcoordinate(i,1);
   Values(s).liney(count,1)=ycoordinate(i,1);
   Values(s).velmag(count,1)=velocitymagnitude(i,1);
   Values(s).xvel(count,1)=xvelocity(i,1);
   Values(s).yvel(count,1)=yvelocity(i,1);

   count=count+1;
 end
 disp(count)
 for j=1:s
    for i=1:lengthx

     maxy=max(Values(j).liney(:,1));

     Values2(j).liney(:,1)=Values(j).liney(:,1)/maxy;
     Values2(j).xvel(:,1)=Values(j).xvel(:,1)/wallspeed;
    end
 end

% %%%%Creating Plots%%%%%

figure(2)

plot(Values2(1,1).xvel(:,1),Values2(1,1).liney(:,1),'-.',...
    Values2(1,2).xvel(:,1),Values2(1,2).liney(:,1),'x',...
    Values2(1,3).xvel(:,1),Values2(1,3).liney(:,1),'*',...
    Values2(1,4).xvel(:,1),Values2(1,4).liney(:,1),'d',...
    Values2(1,5).xvel(:,1),Values2(1,5).liney(:,1),'-',...
    Values2(1,6).xvel(:,1),Values2(1,6).liney(:,1),'s',...
    Values2(1,7).xvel(:,1),Values2(1,7).liney(:,1),'+',...
    Values2(1,8).xvel(:,1),Values2(1,8).liney(:,1),'.')
     title('x-Component of Velocity Magnitude')
```

```
     xlabel('x-Component of Velocity/wallspeed')
     ylabel('y/h')
     legend('Inlet ','7/8 from inlet','3/4 from inlet','5/8 from
inlet',...
          '1/2 from inlet','9/20 from inlet','3/10 from inlet',...
          '1/4 from inlt','location','NorthEastOutside')
       xlim([-0.3,1])

  end
```

Appendix C

Meniscus Shape Factor Derivation

The constant for the meniscus shape is derived within this appendix. The shape factor was created in order to solve the film equations when they reach a pool. At the static pool, the viscous and inertial terms drop out of the equation derived in Chapter 3. It is shown below for convenience.

$$0 = \delta Bo \sin(\alpha)\frac{dh}{dx} + \frac{d\kappa}{dx} - Bo\cos(\alpha) \tag{C.1}$$

Multiply equation C.1 through by the curvature, κ. In order for the equation to equal a constant, it is necessary to integrate the equation. With manipulation of the terms in C.1, the equation can be written as a series of x derivatives. The shape factor listed below is the result of integrating.

$$a = \frac{\delta}{2Bo}\kappa^2 - \frac{\sin(\alpha) - \delta\frac{dh}{dx}\cos(\alpha)}{\sqrt{1+\delta^2\left(\frac{dh}{dx}\right)^2}} \tag{C.2}$$

Where a is a constant. Each constant a corresponds to a different interface shape, and can be tracked to determine where dynamical effects are small compared with static forces. The particular static meniscus of interest describes a pool of liquid for a vertical substrate, so α equals 0. For a pool, the curvature κ goes to 0 as x goes to infinity. In addition, the slope becomes infinite. It follows from equation C.2 that the value of a for static meniscus of interest is one

Appendix D

Full Navier-Stokes Boundary-Layer Equation Derivation

The derivation for the full boundary-layer PDE is outlined within this appendix. The stream function shown ψ below, D.1, is substituted into the boundary-layer approximation in equation set 3.2 in Chapter 3 and with the result shown in equation set D.3.

$$u = \frac{\partial \psi}{\partial y} \qquad v = -\frac{\partial \psi}{dx} \tag{D.1}$$

The kinematic boundary condition used is found in Chapter 3.

$$\xi = \int_0^h \frac{\partial \xi}{\partial y} dy = \psi(h) - \psi(0) \tag{D.2a}$$

The integral solution is simplified by writing the parabolic flow, derived in Appendix C-2, at x equals infinity and solved for the dimensionless flow rate, ξ.

$$\xi = \frac{-1}{3} Bo \cos(\alpha) + 1 \qquad \text{at } x \to \infty \tag{D.2b}$$

Using the solution above, equation D.2b, the kinematic boundary condition simplifies to:

$$C = \psi = 1 - \frac{1}{3} Bo \cos(\alpha) \quad \text{at y=h} \tag{D.3a}$$

$$\psi = 0 \qquad \text{at y=0}$$

The dimensionless groups are found in Chapter 3 as x approaches infinity. The film thickness H equals the final film thickness D leading to the boundary condition shown in D.3b. D.3a shows that continuity is satisfied by using the stream functions.

$$h \to 1 \qquad \text{as} \quad x \to \infty \tag{D.3b}$$

$$\mathrm{Re}\,\delta\left[\frac{\partial \psi}{\partial y}\left(\frac{\partial^2 \psi}{\partial x \partial y}\right) - \frac{\partial \psi}{\partial x}\left(\frac{\partial^2 \psi}{\partial y^2}\right)\right] = -\frac{\partial p}{\partial x} + \frac{\partial^3 \psi}{\partial y^3} - Bo\cos(\alpha) \tag{D.3c}$$

$$\frac{\partial p}{\partial y} = Bo\sin(\alpha) \tag{D.3d}$$

$$\frac{\partial^2 \psi}{\partial x \partial y} - \frac{\partial^2 \psi}{\partial y \partial x} = 0 \tag{D.3e}$$

$$\frac{\partial^2 \psi}{\partial y^2} = 0 \quad \text{at} \quad \text{y=h} \tag{D.3f}$$

$$\text{p} = -\ \kappa \tag{D.3g}$$

$$\frac{\partial \psi}{\partial y} = 1 \qquad \text{at} \quad \text{y=0} \tag{D.3h}$$

$$\psi = 0 \qquad \text{at} \quad \text{y=0}$$

To simplify solving the equation set D.3, it is mapped to a rectangle using the notation below, D.4. After the mapping, the new domain in the y coordinate goes form zero to one.

$$\eta = \frac{y}{h} \tag{D.4}$$

The new coordinate system is given by

$$\psi(x, y) = F(x, \eta)$$

The equations set D.5 is obtained after mapping the equation set D.3 to the new coordinate system.

D.5a

$$\mathrm{Re}\,\delta\left[\left(-\frac{1}{h^3}\right)\frac{dh}{dx}\left(\frac{\partial F}{\partial \eta}\right)^2+\left(\frac{1}{h}\right)\frac{\partial^2 F}{\partial x \partial y}\frac{\partial F}{\partial \eta}-\left(\frac{1}{h^2}\right)\frac{\partial^2 F}{\partial \eta^2}\frac{\partial F}{\partial x}\right]=\frac{\partial \kappa}{\partial x}+\left(\frac{1}{h^3}\right)\frac{\partial^3 F}{\partial \eta^3}+\delta Bo\sin(\alpha)\frac{dh}{dx}-Bo\cos(\alpha)$$

$$\left(\frac{1}{h}\right)\frac{\partial F}{\partial \eta}=1 \qquad \text{at} \quad \eta=0 \qquad \text{(D.5b)}$$

$$\frac{\partial^2 F}{\partial \eta^2}=0 \qquad \text{at} \quad \eta=1 \qquad \text{(D.5c)}$$

$$F=1-Bo\left(\frac{1}{3}\right)\cos(\alpha) \quad \text{at} \quad \eta=1 \qquad \text{(D.5d)}$$

$$h\to 1 \qquad \text{as} \quad x\to\infty \qquad \text{(D.5e)}$$

$$F=Bo\left[\frac{\eta^3}{6}-\frac{\eta^2}{2}\right]\cos(\alpha)+\eta \quad x\to\infty \qquad \text{(D.5f)}$$

The boundary condition D.5f is found by solving for the stream function value as $x\to\infty$. The x-component of the boundary-layer approximation, derived in Appendix C equation C.2a, is simplified as $x\to\infty$ shown below.

$$0 = \frac{\partial^2 u}{\partial y^2} - Bo\cos(\alpha) \tag{D.6a}$$

Equation D.6a is integrated and the integration constant is solved using the shear free boundary condition shown below.

$$\frac{\partial u}{\partial y} = Bo\cos(\alpha)y - Bo\cos(\alpha) \tag{D.6b}$$

Equation D.6b is integrated to solve for the x-component of velocity. The integration constant is solved using the no slip boundary condition at the wall.

$$u = Bo\cos(\alpha)\left[\frac{y^2}{2} - y\right] + 1 \qquad \text{as} \qquad x \to \infty \tag{D.6c}$$

The value of the stream function is found by substituting equation D.1 and equation D.4 solved as $x \to \infty$ the result is shown below.

$$\frac{\partial \psi}{\partial y} = Bo\cos(\alpha)\left[\frac{\eta^2}{2} - \eta\right] + 1 \tag{D.6d}$$

Integrating equation D.6d and solving for the integration constant using the kinematic boundary condition at the wall, D.3a the value of the stream function is found as $x \to \infty$.

$$\psi = Bo\cos(\alpha)\left[\frac{\eta^3}{6} - \frac{\eta}{2}\right] + \eta \qquad \text{as} \quad x \to \infty \tag{D.6e}$$

Appendix E

Plug-Parabola flow Solution MatLab Code

This code solves the third order plug-parabola film equation, 3.5b, by converting it to three first order equations and using ode45. The Rootplug function solves the third order polynomial using Newton's method to get the relaxation parameter m.

E.1 ODE Plug-Parabola Film Equation Solver

```
%ODE Film Equation Solver%
%Plug Flow
%Outputs x,y,z points for slot die Gambit mesh
%Current as of 6-3-10

%--------------------------------------------------------------------%
clc

%Constants
tic
clear all

global Bo Alpha Re We C D W

%Chosen Values for problem
Bo=0.08;                        %Bond Number
Re= 0;                          %Reynolds Number
%Ca=0.1;                        %capillary number
FinalFilm=0.001;                %final Film thickness
C= 1;     %Flow Rate
D=.1; %Ca^(1/3);                %Delta =(capillary number)^1/3
We=Re*D;
Alphadeg= 0;                    %Angle of incline degrees
Alpha=Alphadeg*(pi()/180);      %Convert Alpha to radians

W=1;                            %Constant to solve for L-L or full
                                %Equation can be 0(L-L) or 1

K=.0001;                        %Constant from solving the diff
equation
n=0;                            %Counting Constant
t=15;                           %Value of dh/dx to stop code

  disp(Bo);
  disp(D);

%--------------------------------------------------------------------%

if Bo*cos(Alpha)>1
    disp('Bocos(Alpha) is greater than 1')
```

```
end

%%Root Finder program Newton's Method %%
%F(m)=m^3+m*[Bo*d*sin(Alpha)+ (We*C^3)]+3*(1-Bocos(Alpha))

m=-1*(3*(1-Bo*cos(Alpha)))^(1/3);  %Original guess for root

%Newton's Method Finding Roots
kk=rootplug(m); %Calling function root
F=kk^3+kk*(Bo*D*sin(Alpha)+(Re*D*C^2))-3*(1-Bo*cos(Alpha));
    disp(F);
    disp(kk);
%-------------------------------------------------------------------------%

%%Film Equation Solution%%

%Creating a System of first order ODEs
    %y1=h y2=h' y3=h''
    %System of equations
    %y(1)'=y(2), y(2)'=y(3),
    %y(3)'=[1+d^2*y(2)^2]^(3/2)*{[We]*[-((C/y(1)^3)]*...
    %y(2)+(3*d^2*y(2)*y(3)^2)/[1+d^2*y(2)^2]^(5/2)-
Bo*d*sin(Alpha)*...
    %y(2)-Bo*cos(Alpha)+3*(y(1)-C)/y(1)^3}

while n>=0
y0=[1+K;kk*K;K*kk^2];                    %Initial conditions

xspan=[0 10+n];                          %X interval
maxstep= 0.05;
options=odeset('MaxStep',maxstep);
[x,y]=ode45(@fnplug,xspan,y0,options);    %Diff equation solution


s=length(y);                 %Finding the size of y for loop

%Creating matrix constants
tdriv=zeros(s,1);
crv=zeros(s,1);
dcrv=zeros(s,1);
dstaticm=ones(s,1);
a=zeros(s,1);
aa=zeros(s,1);
h=zeros(s,1);

for i=1:s
%Calculation of d3h/dx3
G =(We)*(((-1*C^2)/y(i,1)^3)*y(i,2));

R =(W*3*D^2*y(i,2)*y(i,3)^2)/((1+D^2*y(i,2)^2)^(5/2));

A = -Bo*D*sin(Alpha)*y(i,2)+Bo*cos(Alpha)-((3*(y(i,1)-C))/y(i,1)^3);

tdriv(i,1)=((1+W*D^2*y(i,2)^2)^(3/2))*(G+R+A);
```

```
%Curvature equation
crv(i,1)=y(i,3)/((1+W*D^2*y(i,2)^2)^(3/2));

%Derivative curvature equation
dcrv(i,1)=(-1*tdriv(i,1)/((1+W*D^2*y(i,2)^2)^(3/2)))+ R;

    if Bo ~= 0
        %Bo does not equal zero

        %Derivative Static meniscus equation
        P=D*y(i,3)*cos(Alpha)*((1+D^2*y(i,2)^2)^.5);

        Q=((sin(Alpha)-D*y(i,2)*cos(Alpha)))*D^2*y(i,3)*-1*...
            y(i,2)*((1+D^2*y(i,2)^2)^(-1/2));

        dstaticm(i,1)=((D/Bo)*crv(i,1)*dcrv(i,1))-((P-
Q)/(1+D^2*y(i,2)^2));

        %Calculation Alpha ()
        aa(i,1)=(sin(Alpha)-
D*y(i,2)*cos(Alpha))/((1+D^2*y(i,2)^2)^(1/2)); %add + from notes
        a(i,1)=((D/(Bo*2))*crv(i,1)^2)- aa(i,1);

    end

%solving of h=1+h'
h(i,1)=1+K*exp(-m*x(i,1));

end

n=n+1; %change time of integration

 %Convergence Check---1st derivative becomes too large

       %Convergence Check---1st derivative becomes too large

        for i=1:s
          v = y(i,2);
            if v > t
                n=-1;
                stoprow=i;
                stop=1;
                break
            end
        end
end
```

```
%-------------------------------------------------------------------------%

%%Creating Plots%%

        %Full equation set
        figure(1)
        plot(x,y(:,1)); %x x-axis
        title('Film Equation Solution')
        xlabel('-x')
        ylabel('y( )')

     if Bo~=0
        figure(2)
        plot(x,a(:,1));
        title('Alpha vs -x')
        xlabel('-x')
        ylabel('Alpha')
        else
            figure(2)
            plot(x,crv);
            title('Curvature vs -x')
            xlabel('-x')
            ylabel('K')
    end

figure(3)
plot(x(1:stoprow,1),y(1:stoprow,2));
title('dh/dx');

figure(4)
plot(x(1:stoprow,1),y(1:stoprow,1),x(1:stoprow,1),y(1:stoprow,2),x(1:st
oprow,1),y(1:stoprow,3)); %x x-axis
title('Film Equation Solution h, 1^s^t and 2^n^d Derivatives ')
xlabel('-x')
ylabel('y( )')
legend('h','dh/dx','d^2h/dx^2')

figure(5)
plot(x(1:stoprow,1),y(1:stoprow,2),x(1:stoprow,1),y(1:stoprow,3)); %x
x-axis
title('Film Equation Solution h, 1^s^t and 2^n^d Derivatives ')
xlabel('-x')
ylabel('y( )')
legend('dh/dx','d^2h/dx^2')

%%-----Points for Meshing of Meniscus shape---------%%
s=length(h);
counth=0;
countstart=0;

for i=1:s
    if y(i,1)<(1+FinalFilm)
        countstart=countstart+1;
    end
```

```
    if y(i,1)<12 %Checking to see if Value of h is great than 10X 1.001
    counth=counth+1;
    end

end

rowstart=countstart+1;

FFD=FinalFilm/D;
Lpoints=counth-rowstart; %finding total number of rows in points

points=zeros(Lpoints+3,3);
rowcount=0;
%Constants to translate x coordinates to have x=0 at inlet

for k=rowstart:counth
rowcount=rowcount+1;
points(rowcount,1)=1000*(x(k,1)*FFD);  %x coordinates
points(rowcount,2)=1000*(y(k,1)*FinalFilm);  % y coordinates
points(rowcount,3)=0.0;     %z coordinates
end

%Adding points for entrance length

     %inlet top right
    points(Lpoints+1,1)=points(Lpoints,1);
    points(Lpoints+1,2)=10;
    points(Lpoints+1,3)=0;

    %inlet top left
    points(Lpoints+2,1)=points(Lpoints,1)+10;
    points(Lpoints+2,2)=10;
    points(Lpoints+2,3)=0;



    %inlet bottom
    points(Lpoints+3,1)=points(Lpoints+2,1);
    points(Lpoints+3,2)=0;
    points(Lpoints+3,3)=0;

    %outlet
    points(Lpoints+4,1)=points(1,1);
    points(Lpoints+4,2)=0;
    points(Lpoints+4,3)=0;

%zeroing points so inlet is at x=0 and shape goes in positive x
direction

pointlength=length(points);
distancex=points(pointlength-1,1)-points(1,1);
setzero=1000*(x(rowstart,1)*FFD);

for u=1:pointlength
    points(u,1)=(-1)*(points(u,1)-distancex-setzero);
```

```
end

 %test does make difference
 points(Lpoints+4,1)=points(pointlength,1)+(5*points(1,2));
    points(Lpoints+4,2)=0;
    points(Lpoints+4,3)=0;

    points(Lpoints+5,1)=points(Lpoints+4,1);
    points(Lpoints+5,2)=points(1,2);
    points(Lpoints+5,3)=0;

    disp(a);
    toc
```

E.2 Rootplug Function

```
function [ m1 ] = rootplug( m )
%%Root Finder program Newton's Method %%
%4-13-10
%ODE solution/Classic Film Equation%
%F(m)=m^3+m*[Bo*d*sin(Alpha)+ (Re*D*C^3)]+3*(1-Bocos(Alpha))

global Bo Alpha Re C D
error= 0.001;        %Setting error constant

    while error>1e-12
    %m1 is new guess from solving m
        F=m^3+m*(Bo*D*sin(Alpha)+(Re*D*C^2))...
            -3*(1-Bo*cos(Alpha));              %Guess plugged in F(m)

         Fp=3*m^2+(Bo*D*sin(Alpha)+(Re*D*C^2)); %F'(m) with guess

        m1=m-(F/Fp);                  % Finding new root
        error=abs(1-(m/m1));          % Finding Error
        m=m1;                         % Replacing Old Guess
    end

disp('Root By Newton''s Method')
disp( num2str(m1))
end
```

E.3 fnplug Function

```
function dhdx=fnplug(x,y)
%6-3-10
global Bo Alpha We C D W Re
 %Equations in from of 3 separate equations for ODE solver

 dhdx=[y(2);y(3);((1+W*D^2*y(2)^2)^(3/2))*(((Re*D*(-
1)*C^2*y(2))/y(1)^3)-...
    Bo*D*sin(Alpha)*y(2)+((3*(y(1)-C))/y(1)^3)+((3*D^2*y(2)*...
    y(3)^2)/(1+D^2*y(2)^2)^(5/2))-Bo*cos(Alpha))]; %NEG x

 end
```

Appendix F

Parabola Film Equation Solution MatLab Code

This code solves the third order parabola film equation, 3.4b, by converting it to three first order equations and using ode45. The Root function solves the third order polynomial using Newton's method to get the relaxation parameter m.

F.1 ODE Parabola Film Equation Solver

```
%ODE Film Equation Solver%
%Outputs x,y,z points for slot die Gambit mesh
%Current as of 2-5-10

%-------------------------------------------------------------------------%
clc
%Constants

global Bo Alpha Re C D W

%Chosen Values for problem
Bo= 1;                                  %Bond Number
Re= 0;                                  %Reynolds Number
Ca=0.1;                                 %Capillary number
FinalFilm=0.001;                        %final Film thickness
C= 1;
D= Ca^(1/3);                    %Delta =(capillary number)^1/3

Alphadeg= 0;                    %Angle of incline degrees
Alpha=Alphadeg*(pi()/180);      %Convert Alpha to radians

W=1;                            %Constant to solve for L-L or full
                                %Equation can be 0(L-L) or 1
K=.0001;                        %Constant from solving the diff
equation
n=0;                            %Counting Constant
t=100;                          %Value of dh/dx to stop code

  disp(Bo);
  disp(D);

%-------------------------------------------------------------------------%
if Bo*cos(Alpha)>1
    disp('Bocos(Alpha) is greater than 1')
end



%%Root Finder program Newton's Method %%
```

```
%F(m)=m^3+m*[Bo*d*sin(Alpha)-(Re*d/5)*(1-6C^2)]+3*(1-Bocos(Alpha))

m=-1*(3*(1-Bo*cos(Alpha)))^(1/3);  %Original guess for root

%Newton's Method Finding Roots

kk=root(m);                          %Calling function root
F=kk^3+kk*(Bo*D*sin(Alpha)-((Re*D/5)*...
        (1-6*C^2)))+3*(1-Bo*cos(Alpha));
    disp(F);

%-------------------------Film Equation Solution--------------------%
%Creating a System of first order ODEs
    %y1=h y2=h' y3=h''
    %System of equations
    %y(1)'=y(2), y(2)'=y(3),
    %y(3)'=[1+d^2*y(2)^2]^(3/2)*{[Re*d/5*y(1)]*[1-(6*C^2/y(1)^2)]*...
    %y(2)+(3*d^2*y(2)*y(3)^2)/[1+d^2*y(2)^2]^(5/2)-Bo*d*sin(Alpha)*...
    %y(2)-Bo*cos(Alpha)+3*(y(1)-C)/y(1)^3}

while n>=0
y0=[1+K;-m*K;K*m^2];         %Initial conditions
xspan=[0 10+n];
[x,y]=ode45(@fn,xspan,y0);   %Diff equation solution

s=length(y);                 %Finding the size of y for loop

%Creating matrix constants
tdriv=zeros(s,1);
crv=zeros(s,1);
dcrv=zeros(s,1);
dstaticm=ones(s,1);
a=zeros(s,1);
aa=zeros(s,1);
h=zeros(s,1);

for i=1:s
%Calculation of d3h/dx3
G =(Re*D/5*y(i,1))*((1-(6*C^2/y(i,1)^2))*y(i,2));

R =(W*3*D^2*y(i,2)*y(i,3)^2)/((1+D^2*y(i,2)^2)^(5/2));

A = -Bo*D*sin(Alpha)*y(i,2)-Bo*cos(Alpha)+((3*(y(i,1)-C))/y(i,1)^3);

tdriv(i,1)=((1+W*D^2*y(i,2)^2)^(3/2))*(G+R+A);

%Curvature equation
crv(i,1)=y(i,3)/((1+W*D^2*y(i,2)^2)^(3/2));

%Derivative curvature equation
dcrv(i,1)=(-1*tdriv(i,1)/((1+W*D^2*y(i,2)^2)^(3/2)))+ R;

    if Bo ~= 0
        %Bo does not equal zero
```

```
        %Derivative Static meniscus equation
        P=D*y(i,3)*cos(Alpha)*((1+D^2*y(i,2)^2)^.5);

        Q=((sin(Alpha)-D*y(i,2)*cos(Alpha)))*D^2*y(i,3)*-1*...
            y(i,2)*((1+D^2*y(i,2)^2)^(-1/2));

        dstaticm(i,1)=((D/Bo)*crv(i,1)*dcrv(i,1))-((P-
Q)/(1+D^2*y(i,2)^2));

        %Calculation Alpha ()
        aa(i,1)=(sin(Alpha)-
D*y(i,2)*cos(Alpha))/((1+D^2*y(i,2)^2)^(1/2));
        a(i,1)=((D/Bo*2)*crv(i,1)^2)- aa(i,1);

    end

%solving of h=1+h'
h(i,1)=1+K*exp(-m*x(i,1));

end

n=n+1; %change time of integration

 %Convergence Check---1st derivative becomes too large

       %Convergence Check---1st derivative becomes too large
     for i=i:s
        v = y(i,2);
         if  v > t
             if Bo~=0
                  %Checking the 1st derivative value
                 if abs(dstaticm(i,1)) < 0.001
                        n=-1;       %Value of n to break while loop
                   break             %Breaks for Loop
                 end
             else
                 %Case when Bo=0
                 if dcrv(i,1) <.001
                     n=-1;
                     break
                 end
             end
         end
     end

             if n>300
                 n=-1;
             end
end


%%------Creating Plots and Points for Meshing of Meniscus shape------%%
```

```
s=length(h);
counth=0;
countstart=0;

for i=1:s
    if y(i,1)<1.001
        countstart=countstart+1;
    end

    if y(i,1)<12 %Checking to see if Value of h is great than 10X 1.001
    counth=counth+1;
    end
end
```

F.2 Root Function

```
function [ m1 ] = root( m )
%%Root Finder program Newton's Method %%
%1-8-10
%ODE solution/Classic Film Equation%
%F(m)=m^3+m*[Bo*d*sin(Alpha)-(Re*d/5)*(1-6C^2)]+3*(1-Bocos(Alpha))

global Bo Alpha Re C D
error= 0.001;        %Setting error constant

    while error>1e-12
    %m1 is new guess from solving m
        F=m^3+m*(Bo*D*sin(Alpha)-((Re*D/5)*...
            (1-6*C^2)))+3*(1-Bo*cos(Alpha)); %Guess plugged in F(m)

        %F'(m) with guess
        Fp=3*m^2+(Bo*D*sin(Alpha)-((Re*D/5)*(1-6*C^2)));
        m1=m-(F/Fp);                    % Finding new root
        error=abs(1-(m/m1));            % Finding Error
        m=m1;                           % Replacing Old Guess
    end

disp('Root By Newton''s Method')
disp( num2str(m1))
end
```

F.3 fn Function

```
function dhdx = fn(x,y)
%1-8-10
global Bo Alpha Re C D W

%Equations in from of 3 separate equations for ODE solver
 dhdx=[y(2);y(3);((1+W*D^2*y(2)^2)^(3/2))*((Re*D/5*y(1))*...
    (1-(6*C^2/y(1)^2))*y(2)+(W*3*D^2*y(2)*y(3)^2)...
    /((1+D^2*y(2)^2)^(5/2))- Bo*D*sin(Alpha)*y(2)- Bo*cos(Alpha)...
    +(3*(y(1)-C))/y(1)^3)];

 end
```

Appendix G

MatLab Code for Importing Fluent Data

This code is used to import the data exported from Fluent. The file must be entered into the function as follows Fluent ('filename.cvs'). This code imports the data into different arrays. There is no separation between the data and between the different lines; and therefore, requires manipulation in order to plot the different sets of data.

```
function Fluent(openfile,varargin)
%
% FLUENT ( 'OPEN FILE' )
%
% Read a .CSV output file from Fluent which contains both text and
% numerical data. Variables are created to store numerical data in the
form
% of row vectors according to standard column headers of Fluent ASCII
% output. These variables are assigned to the workspace for immediate
use
%
% OPEN FILE: string containing the path of the data file to be read
%
% This function expects the file to have one header row containing
column
% headers and all subsequent rows to contain the corresponding data of
each
% header.
%
% CSV read algorithm: Rebecca Jaiven & William Seely, General Electric
% Comp.
%
% Revision 0.1: Remove sections which would allow the user to chose
% 'OPEN FILE' and 'SAVE FILE' during operation rather than specify
these as
% function arguments.
%
% Revision 0.2: Remove options to save .MAT file
%
%
% Revision 0.2
% Mark Livelli
% Rochester Institute of Technology
% March 24, 2009
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Set Constants
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

% Set file delimiter
delim = ',';

% Set format placeholder
dataformat = [];

% Set data file
openfile; %#ok<VUNUS>

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load File
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Open file
fid = fopen(openfile,'r');

% Display message if file does not exist
if fid == -1
    error('File Not Found')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Prepare Data Formats
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Read file header
headerstring = fgetl(fid);

% Parse file header
headercell = textscan(headerstring,'%s','delimiter',delim);
headercell = headercell{1};

% Parse data
data = textscan(fid,'%s',length(headerstring),'delimiter',delim);
data = data{1};

% Construct column formats
for i = 1:length(data)

    % If number
    if ~isnan(str2double(data{i}))
        columnformat = '%f ';

    % If string
    else
        columnformat = '%s ';
```

```
    % Store format
    end
    dataformat = [dataformat columnformat]; %#ok<AGROW>
end

% Return to start of file
frewind(fid)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Create Data Fields
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Read file data with column formats
data = textscan(fid,dataformat,'delimiter',delim,'headerlines',1);

% Close file
fclose(fid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Assign Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Assign data to variables of corresponding header in workspace
for i = 1:length(headercell)

    % Retrieve variable name
    variable = headercell{i};

    % Ensure acceptable format
    variable = strrep(variable,'-','');
    variable = lower(genvarname(variable));

    % Replace improperly formatted
    headercell{i} = variable;

    % Assign data
    assignin('base',variable,data{i});
    eval([variable '=data{i}' ';'])
end
```